Who Actually Built Your Open Model? Soofi S vs Trinity

August 25, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** Two labs shipped competitive open mixture-of-experts models this year with no hyperscaler behind either one. I went and read both technical reports to see what that independence actually bought — and found that one of them declares another company's architecture string in its own file header.

Three weeks ago I was going to write that the open mixture-of-experts tier is exclusively corporate. Every MoE that makes local inference practical came from a company: Qwen from Alibaba, Gemma from Google, GLM from Z.ai, DeepSeek from High-Flyer, Kimi from Moonshot. The argument was that reproducible MoE training costs more than dense training, because you have to publish routing ablations and expert-utilisation logs and not just weights, so no independent lab could afford it.

I checked before writing, and the claim died on two counterexamples. Arcee AI, a San Francisco startup, had shipped the Trinity family under a permissive licence. The Soofi project, a German publicly funded consortium, had shipped a 30B-A3B base model with per-source token accounting. Both trained from scratch. Both are independent in the sense that matters.

So I read both technical reports properly to find out what that independence consists of. The answer is more interesting than the thesis it replaced, and it is not the cynical one.

The correction first

The old framing was wrong in a specific way worth naming: it treated “who trains the model” as the whole question. Once you look at what a lab has to assemble before it can train anything, that question turns out to be one layer of several, and the labs that pass the independence test at the top can differ enormously underneath.

That is the finding. Not “independence is fake”, because it isn’t. Trinity’s weights are out under a permissive licence and nobody can withdraw them. The finding is that independence at the lab level tells you much less than I assumed about what the model is made of.

Soofi S: sovereign at the top, NVIDIA most of the way down

Soofi S is the more striking case because its entire premise is reducing dependence. It is coordinated by the German AI Association with Fraunhofer IAIS and IIS, DFKI, TU Darmstadt, the University of Würzburg, Leibniz University Hannover, Berlin University of Applied Sciences, ellamind and Merantix Momentum, on federal ministry funding. The paper's title is A Sovereign, Open-Source Foundation Model for German and English.

Its technical report is unusually candid about what it is built on. Four separate layers trace back to one vendor.

The architecture is NVIDIA's, unmodified, and the paper says so in those words. Soofi S adopts the Nemotron 3 Nano reference design — a 52-layer hybrid interleaving 23 Mamba-2 sequence-mixing layers, 23 MoE layers and 6 grouped-query attention layers, with 128 routed experts plus 2 shared, 6 active per token. The report states that because it reuses the reference design without modification, it refers readers to NVIDIA's own Nemotron reports for the design motivation and the ablations behind each architectural choice. Table 1 is captioned as following the Nemotron 3 Nano reference configuration. This is not a case of a paper being caught out; it is a paper being straightforward.

The data is largely NVIDIA's. The report anchors its mixture on the openly released Nemotron-CC datasets: v2.1, v2.0 and v1.0 contributing roughly 2.84T, 5.60T and 3.15T effective tokens respectively. Code adds about 3.38T across NVIDIA's Nemotron code datasets. Specialised STEM comes from Nemotron-Pretraining-Specialized-v1 at about 1.35T. Mathematics contributes roughly 1.06T from Nemotron-CC-Math. The German-language component is genuinely the consortium's own work, about 1.65T effective tokens, 7.2% of phase one against the 5% multilingual share of NVIDIA's reference mixture, and that is the part of the corpus the project actually built.

The tokenizer is NVIDIA's. The report notes that its exact per-iteration token counts were obtained by tokenizing with the Nemotron-3 tokenizer. The vocabulary that decides how German words get split is not the consortium's.

The silicon is NVIDIA's, and so is the sovereign cloud. Training ran on the Industrial AI Cloud in Munich, operated by Deutsche Telekom, using up to 512 NVIDIA B200 GPUs across 64 DGX B200 nodes, consuming roughly 253,000 B200 GPU-hours between 24 March and 13 May 2026. The report describes running on German soil under European operational and data-protection requirements as part of the model's sovereignty — and in the same passage notes that the facility was built together with NVIDIA.

I want to be careful here, because there is a cheap version of this observation and it is wrong. Nemotron-CC is openly released. The Nemotron 3 Nano design is openly published. Reusing them is legitimate engineering and the report argues the case explicitly: the architecture is already integrated into the major open inference and serving stacks, which makes the model practical to deploy on day one, and sharing a backbone with a known model turns it into a scientific control. Those are good reasons. The consortium is not hiding anything — I know all of this because they wrote it down.

The point is narrower, and it is about which decisions a lab still controls. A model whose architecture, data pipeline and tokenizer all originate with one company inherits that company's design decisions wholesale, and a sovereignty argument that stops at the datacentre door does not reach any of them.

The part that shows up in the metadata

Here is where this stops being a paper-reading exercise.

The Soofi S GGUF repository declares its architecture as `nemotron_h_moe`. That is NVIDIA's architecture string, sitting in a German sovereign model's metadata. It is there because the conversion is honest: llama.cpp has to know which code path to run, and the code path is the Nemotron one.

I want to be exact about what I checked, because this is the claim a reader is most likely to try to reproduce. That string comes from the repository's own model card, not from a header I opened. The repo is gated: a range request against the GGUF returns HTTP 401 and the message that access to the model is restricted. So I could not read the file, and neither can you without being granted access.

The same card describes the checkpoint as a custom hybrid Mamba-2/MoE model and never mentions Nemotron anywhere in its prose. That gap is the observation. The dependency is stated in the machine-readable field and absent from the sentence a human reads, and the file where you would confirm it for yourself sits behind a gate.

Trinity I could check directly, because it is ungated. This is the header on the Trinity-Mini file sitting on my own bench box, read with a GGUF parser rather than taken from a card:

```
general.architecture      = afmoe
general.basename          = Trinity
general.base_model.0.name = Trinity Mini PreRL Fusion
afmoe.expert_count        = 128
afmoe.expert_used_count   = 8
```

`afmoe` is Arcee's own architecture family, registered in llama.cpp under its own name. Same field, same tooling, different answer.

For any model you can actually download, that field is the cheapest provenance check there is. It will not tell you where the data came from and it will not tell you who owns the tokenizer. It will tell you whose design the code is executing, and it takes one command. Both `nemotron_h_moe` and `afmoe` are recognised architectures in the llama.cpp build we pin for benchmarks, so neither is exotic.

Trinity: independent in a different shape

This is where the thesis I set out to write needed narrowing, because the two cases do not match.

Arcee's Trinity Large technical report describes an architecture the team specified themselves — an extremely sparse MoE with interleaved local and global attention plus gated attention. It cites DeepSeek, GLM, Kimi and Qwen as influences, which is what published research is for, but it does not adopt a reference configuration wholesale. Trinity Large is 398B total with roughly 13B active; Trinity Mini, the one in our bench tier, is 26B with about 3B active.

The tokenizer is theirs: a custom 200,000-token BPE vocabulary trained from their own corpus. The pretokenizer pipeline is openly credited as inspired primarily by DeepSeek V3's, and they adopt DeepSeek's main text regex directly — so there is borrowing here too, but of a published technique rather than a delivered artifact.

The data was curated by DatologyAI, and the compute ran on clusters managed by Prime Intellect: 512 H200s for Nano and Mini, 2,048 B300s for Trinity Large. Both organisations are co-authors on the report rather than distant suppliers.

So Trinity is not self-sufficient either. Nobody is. But its dependencies point at three different parties, none of which supplies more than one layer, and the silicon vendor supplies only silicon. Soofi S's dependencies converge on one company across four layers.

That difference is the actual result, and it is the opposite of what I expected going in. The publicly funded sovereignty project is the more vertically dependent of the two. The venture-backed American startup is the more architecturally independent.

The awkward licensing footnote

One thing has to be said plainly because it cuts against Soofi S and I would be shading the piece to leave it out.

As of today, Soofi S is not actually an open release. The base model card gates access behind sharing your contact details, describes the checkpoint as a beta preview and research artifact that is not an open release, and says it is in closed beta with already selected partners. The licence field says only that a permissive licence will accompany the final release.

Trinity is ungated today. Its model cards carry OpenMDW-1.1, the Linux Foundation's permissive model licence, whose only substantive condition is that redistribution preserves the licence and its notices. Worth noting that launch coverage in late 2025 described Trinity as Apache 2.0; OpenMDW-1.1 did not exist until 28 May 2026, so the cards appear to have been relicensed since. Either way it is permissive, and either way you can download it right now. There is a small irony in the licence: NVIDIA adopted OpenMDW for the Nemotron family too, so the two models in this piece share a licence even where they share nothing else.

So one of the two counterexamples that killed my original thesis is, at this moment, a model most people cannot download. That does not resurrect the thesis — the training happened, the report is public, the per-source accounting is real and roughly 99% of the mixture is independently reconstructible, which is more openness than most corporate releases offer. But if you are choosing something to run this week, only one of these two is a candidate.

What this means if you are picking a model to run

The practical version, for the audience this site actually has.

Architecture provenance predicts your tooling experience. Soofi S runs on day one in anything that already supports Nemotron 3 Nano, which is most of the open serving stack. That is a genuine, immediate benefit of reusing a reference design, and the report names it as a reason. Trinity needed `afmoe` support added. Borrowed architecture is faster to deploy; the cost is that your model inherits someone else's ceiling.

Tokenizer provenance predicts how your language gets handled. Soofi S up-weighted German data heavily but counts it with NVIDIA's tokenizer. Trinity trained its own vocabulary and reports competitive English and French compression while trailing DeepSeek V3 and Qwen 3 on CJK, which they attribute to the timing of their training data. If you work in a non-English language, the vocabulary is a real variable and it is rarely discussed.

Data provenance predicts what the model has seen. Four Nemotron-derived corpora account for the bulk of Soofi S's non-German tokens. Whatever is systematically absent from Nemotron-CC is systematically absent from Soofi S.

None of that makes either model a bad choice. It makes them different objects, and the differences are documented in both cases if you go and read.

Two data points, and I am not going to pretend otherwise

This is two labs. It is not a survey and I have not established a pattern.

What I can say is that the two cases sit at opposite ends of a range, which at least tells you the range exists. A lab can be independent at the top and near-totally dependent underneath, or independent at the top and dependent on several unrelated parties underneath, and both configurations currently ship models people describe with the same word.

The honest generalisation is smaller than the one I originally wrote in my notes. "The dependency moved down the stack" is exactly right for Soofi S. It is only partly right for Trinity, and asserting it as a general law of independent labs would repeat the mistake that killed the first version of this piece — reaching for a clean claim before checking whether the cases actually match.

If more independent MoE labs ship, this is worth revisiting with a real sample. Until then, two.

The bottom line

Independent labs can train competitive open MoE models. That question is settled and I was wrong to doubt it.

What independence buys is narrower than the word suggests. It reliably buys you control of the weights and the licence, which is the part that matters for whether you can run the thing forever without permission. It does not automatically buy you control of the architecture, the corpus, the vocabulary or the hardware, and one of these two projects gave up all four to one vendor while describing itself as sovereign.

The cheapest thing you can do about any of this is read the header. `general.architecture` costs one command and tells you whose design you are running. It is a small check, it is more reliable than the prose on a model card, and the one model here where I could not run it is also the one you cannot download.

Related guides

- China Made Open Source a Strategy. If It Pulls Back, Who Fills the Gap? — the geopolitical frame this sits inside, and why Western open releases matter
- Open Weights You Can't Run — the other limit on open models: published, licensed, and too large for your hardware
- MoE Models Explained — what routed experts are and why the architecture makes local inference viable
- Is Qwen Going Closed? — the openness question from the corporate side

Get notified when we publish new guides.

Subscribe — free, no spam
