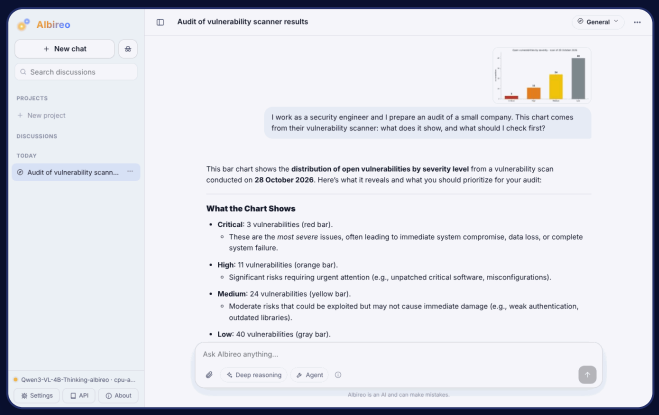
Albireo

Your private AI assistant, running on your own computer

Albireo runs language models directly on your Windows 11 computer. Your conversations, documents and models stay on the machine: no account to create, nothing sent outside.

Version **1.0.23** Platform **Windows 11, x64**

License **Boost Software License 1.0**



56.8
tokens per second on a laptop GPU, against 23.6 on its processor

22
models in the signed catalog, from 7 publishers

5
engines: processors, Intel NPU and GPU, NVIDIA GPU

0
account, subscription or data sent outside

At a glance

Chat
Discussions kept in the browser, an incognito mode, a search across the discussions, and the reasoning of the models that reason shown while they think.

Images
The models that read images take PNG and JPEG pictures, attached, pasted or dropped into a question.

Agent mode
The model calls tools that run in the page: calculator, sandboxed JavaScript, reading and search of the documents, up to 100 steps per answer.

Reading aloud
Each answer can be read aloud by the speech synthesis that comes with Albireo, without network.

Offline mode
On a computer without network, the service opens no connection: the catalog, the models and the updates come from a signed bundle brought on a disk.

Documents
Word, Excel, PowerPoint, OpenDocument, PDF, RTF, HTML, text, CSV, Markdown and source code, read in the browser and attached to a message or a project.

Projects
Discussions grouped with instructions and documents that form their shared context, searched by meaning and by words.

Web access
An optional search of Wikipedia and the reading of public pages and PDF files, off in each new discussion.

An interface for other programs
Chat completions, completions, embeddings, speech and models follow the OpenAI interface, with streaming and images, documented in a Swagger page.

Relay to an external engine
A discussion can be relayed to an external engine: any interface compatible with OpenAI, the cloud of Ollama or Claude. Its texts are pseudonymized by Pseudalis before they leave the computer, and an engine of the computer or of the local network remains usable offline.

Engines

Engine	Runs on
`cpu`	Every x86-64 processor
`cpu-avx2`	Processors with AVX2, FMA and F16C: Intel Haswell, AMD Zen and later
`cpu-avxvnni`	Processors with AVX-VNNI: Intel Core from Alder Lake, Intel Xeon from Sapphire Rapids, AMD Zen 5
`openvino`	NPU and integrated GPU of Intel processors, through OpenVINO
`cuda`	NVIDIA graphics processors, through CUDA

Speed on everyday hardware: the techniques of Albireo

Albireo runs models with its own engine, designed for the processors and graphics cards already in service. Each technique below is measured in the README of the product.

Quantized kernels

The weights stay in the memory mapping of the GGUF file, in their quantized form. Integer kernels compute 32 products where a vector of floats computes 8.

Reading a prompt: 216 tokens per second for Qwen3-VL 4B and more than 1,300 for Qwen3 0.6B (Intel Core Ultra 9 275HX, 24 threads); on two cores, 230 against 126 on floats (Qwen3 0.6B, AVX-VNNI)

An engine for each kind of processor

Portable, AVX2 and AVX-VNNI engines for processors, the NPU and the integrated GPU of Intel through OpenVINO, NVIDIA graphics processors through CUDA. The service measures them on the machine and runs the fastest.

Writing: 80.6 to 87.7 tokens per second for Qwen3 0.6B with the processor engine (Intel Core Ultra 9 275HX, 24 threads)

V²RAM extended memory

The extended device memory of the library lets the copies of the weights go beyond the memory of the card, into host memory or a file, and brings them back ahead of their use.

Several conversations at once

The requests of every client are computed together in each forward pass, and the keys and values of a finished request stay cached for the next turn.

32 to 40% less time per step with four answers written together

Drafted tokens

An answer writes several tokens per step when its context foretells them, a form of speculative decoding. The model keeps the tokens it would have drawn itself, so that the answers keep the distribution of the sampling.

25.5 against 19.7 tokens per second when rewriting code (Qwen3 0.6B, two cores); 26.3 against 23.6 on 24 threads (Qwen3-VL 4B)

Layers computed on the NVIDIA card

The CUDA engine keeps the weights on the card in their GGUF formats and computes the layers of a dense model there, attention included for the generated tokens.

56.8 against 23.6 tokens per second (Qwen3-VL 4B, RTX 5070 Ti Laptop GPU against 24 threads)

Compact conversation memory

Keys and values are kept in 8 bits, in pages shared between the conversations that start with the same text.

34 bytes against 64 per group of values, with the same next token on 210 positions

An OpenVINO graph on the Intel processor and GPU

With OpenVINO, a whole model is computed as a single graph on the processor or on the integrated GPU of Intel, and the NPU takes matrix products. The graph reads the first pass of a prompt faster; the processor engine stays faster on the following passes and when writing, and the measure of the engines, started from the tray icon, keeps the best choice for the machine.

First reading of a prompt: 260 tokens per second with the graph against 216 with the processor engine (Qwen3-VL 4B, Intel Core Ultra 9 275HX, 24 threads)

Characteristics

Model format	Quantized GGUF: Q4_0 to Q8_0, Q2_K to Q6_K and IQ4_NL
Architectures	llama, mistral, mistral3, granite, granitemoe, phi3, qwen2, qwen3, qwen3moe, qwen3vl, qwen3vlmoe, mixtral
Image encoders	Qwen3-VL, LLaVA-NeXT (EuroVLM) and Pixtral (Ministral 3, Devstral Small 2, Mistral Small 3.2)
Models that come with Albireo	Qwen3-VL-4B-Instruct with its image encoder (2.95 GB) and Qwen3-Embedding-0.6B (0.64 GB)
Catalog	22 models from 7 publishers, signed and checked against their hashes
Memory	8 GB for the model installed by default
Local interface	https://localhost:8443, compatible with the OpenAI interface, documented with Swagger
Conversation formats	ChatML, Mistral, Llama 3, Granite, Phi-4-mini and Foundation-Sec, written natively
Updates	Signed feed, installers checked by size and SHA-256

Security you can audit

Local only

The service listens on 127.0.0.1 and refuses the requests whose host or origin is not local.

A certificate authority of the computer

The HTTPS certificate comes from an authority made on the computer with an ECDSA P-256 key, restricted by name constraints to localhost.

Keys protected by Windows

The private key of the server and the key of the relay are encrypted with DPAPI and never written in clear.

A signed supply chain

The update feed and the model catalog are signed with ECDSA P-256, and every downloaded file is checked against its SHA-256.

An open interface

The routes follow the OpenAI interface and are described with OpenAPI, so that existing client libraries work unchanged.

Isolated speech synthesis

The synthesizer runs in a separate process of low integrity, without privileges and bounded in memory.

Free and open source under the Boost Software License 1.0. © 2026 Cydral Technology, France.