

MODULES

In-app AI assistant

Choosing a provider (cloud and local LLMs), permissions and consent, network diagnostics, SCADA screen design, attachments, skills, context management.

Online: mergenhub.envest.com.tr/docs/ai-assistant

ON THIS PAGE

Choosing a provider	Data and privacy
Behaviour settings	Data processing consent (GDPR / KVKK)
What the assistant does	Prices, cost and model info
Permissions and consent	Approval modes
Conversations, context and skills	Scripts, seeing the screen, and permissions

MergenHub ships an in-app AI assistant that works **with the permissions of the signed-in user**. Open it with the  button in the top bar or Plugins › **AI Assistant**: it docks on the right side of the application and can be popped out into a separate window (a native second window in the desktop app). External assistants connect through the [MCP server](#) — both use the same tool catalogue and the same authorization checks.

Choosing a provider

An administrator (Settings permission) defines one or more model providers under  › **Providers** in the assistant panel:

- **Cloud:** OpenAI (ChatGPT), Anthropic (Claude), OpenRouter, Google Gemini, Mistral, DeepSeek, Groq, xAI, Together, Fireworks, Perplexity, Cerebras, Cohere, Azure OpenAI.
- **Local (no key):** Ollama, LM Studio, vLLM, llama.cpp server (Unsloth GGUF), LocalAI, Jan, text-generation-webui, KoboldCpp, LiteLLM/OpenWebUI proxy. **"Scan local servers"** finds running ones and adds a profile with one click.
- **Custom:** any OpenAI-compatible or Anthropic-compatible endpoint.

Each profile has a base URL, an API key (stored only on the server, returned masked), a default model ("Fetch models" lists them), a context window (manual when the API does not report it) and enabled/default flags. The model chip in the chat header switches profile and model from a **searchable list**; the choice is saved per conversation.

If no provider exists yet, the panel shows the administrator a **3-step wizard** (provider → key → model, test). When the assistant is disabled the menu entry is dimmed and clicking it explains where to enable it.

Behaviour settings

- **Assistant enabled** — when off, nobody can chat.
- **Assistant may make changes** — off: read-only tools; on: configuration/alarm/logging/SCADA write tools **within the user's own permissions**, every write audited as `user (AI assistant)`. There is **never** a tool that writes live values to a device.
- **Reasoning / thinking** (off/low/medium/high) — supported by OpenAI o-series and GPT-5, Claude (extended thinking), Gemini 2.5, OpenRouter, Groq, DeepSeek-R1; the thinking stream appears as a collapsible block.
- Temperature, max output tokens, tool steps per turn, **administrator instructions** (site-specific rules).

What the assistant does

It works like a SCADA/automation engineer: read first, write precisely, then verify.

- **Inventory and diagnostics:** channels/devices/tags, live values and quality, device status, alarms, historian, events, audit log, users and permissions, settings — everything your permissions allow.
- **Network tests:** `ping`, TCP port checks (Modbus 502, S7 102, OPC UA 4840...), DNS resolution, local listening ports. "The device is not communicating" is answered by measurement, not guesswork.
- **Configuration:** create/update/delete channels, devices, tags, virtual tags (expression validated), aliases, alarms (ISA-18.2), log groups, Data Bridge agents.
- **SCADA screens:** real symbol library, pipelines with flow animation, tank/pump scenes, bindings, previews. The built-in "**SCADA screen design**" skill provides the layout grid, ISA-101 colour language, navigation and validation rules.
- **System:** partial settings updates (OPC UA port, HTTPS...), HTTPS certificate (self-signed or a PFX attached to the chat), Windows Firewall ports, service restart. Example: "my domain is X, publish HTTPS on 443" → DNS → is the port free → certificate → settings → firewall → restart → verification.
- **Attachments:** images (PNG/JPG...), PDF, CSV/JSON/TXT, SVG and PFX can be attached; images go to vision-capable models and can be placed on screens with `add_widget type=image`, CSV/JSON content becomes context.

`Channel.Device.Tag` references in answers are clickable and navigate the main application. Tool calls appear as cards with arguments, results and previews; a status line ("Thinking / Creating screen / Testing ports") shows where the work is.

Permissions and consent

Security layers: (1) the "**Use the AI assistant**" permission — granted per user/group, without it the assistant is off; (2) every tool call is limited to the user's **own permissions**; (3) the administrator's **write gate**; (4) **approval** — by default every change waits for Approve/Reject in the chat, the assistant never acts unasked; (5) **audit log** — every tool call (reads included) and chat message is logged; (6) there is no live-value write tool.

Token economy: the model receives the core tool groups plus a compact catalogue, not all tools every turn; required groups are loaded during the conversation. Templates: say "make this device type a template" and the assistant saves it as a skill; afterwards one sentence creates channel/device/tags/alarms/screen of that type.

The assistant uses no token: it is the **signed-in user** and acts with that user's permissions. If a tool is refused because of a missing permission (e.g. ConfigWrite) or the write gate, the assistant states the reason and asks; an administrator can open the gate from the card with "**Grant and retry**". No workarounds, no guessing.

Conversations, context and skills

- Conversations are private to the user; rename, folders, pinning, archive and search are available.
- The **context meter** compares the last turn's tokens with the model's window; at 75 % older turns are summarised automatically (the full history stays on screen). Click the meter to compact manually.
- **Skills** (`DataRoot/ai/skills/*.md`) are site-specific instruction packs. Two built-in expert skills ship enabled (SCADA screen design; alarm/historian/bridge/virtual-tag engineering); administrators edit, disable or add their own. Only name and description enter the system prompt; the body is loaded on demand.
- **External MCP servers:** tools from an ERP, a document store or another MergenHub can be added (Streamable HTTP + optional bearer token).

Data and privacy

Conversations, attachments and skills live in the MergenHub data directory (`ai/`); API keys are stored only in `settings.json` and returned masked. Only the text/images needed for the current turn are sent to the provider; with a local provider (Ollama, LM Studio...) nothing leaves the machine.

Data processing consent (GDPR / KVKK)

Before a cloud provider (OpenAI, Anthropic, OpenRouter, Google...) can be used, an administrator must give **data processing consent**; without it no chat opens on a cloud profile. The text covers: you are the data controller and the provider is a third party; the data transferred (messages, attachments, tool outputs: project configuration, tag values, alarm/event records, user names — keys and passwords are masked); the legal basis for transfers abroad; retention; informing your users; security; and the record of who accepted and when, with revocation. Local providers (Ollama, LM Studio, vLLM...) keep data on the machine; no consent needed.

Prices, cost and model info

As soon as an API key is entered the model list is fetched automatically; each model shows its context window and price (per 1M tokens in/out). Prices come from the public OpenRouter catalogue (24 h cache), local models count as free; for unknown models only token counts are shown. The estimated cost of each turn and of the conversation appears under the chat and in the conversation list. The model can be changed mid-conversation; the "Thinking" chip selects the reasoning level per conversation.

Approval modes

Three modes for tools that change things: **Ask on every change** (default), **Remember a tool approved once in the conversation**, **Apply without asking** (administrators only; for closed environments). The approval card offers "do not ask again in this conversation" and "always allow"; permanent allowances are listed and removable under Settings › Behaviour. Every tool call is audited in all modes.

Scripts, seeing the screen, and permissions

The assistant can run Python or PowerShell scripts in a **sandbox** for calculations, reports and conversions. The feature is **off by default** and only the **system administrator (Root)** can enable it: Settings › AI › Behaviour › *Script execution*. A risk statement must be read and accepted first; the acceptance (who and when) is written to the immutable compliance ledger.

How scripts are contained:

- They run as a separate **low-privilege Windows account** (MergenHubSandbox) and cannot read the MergenHub data folder.
- **Network access is off by default.** If you enable it you provide an allowlist (IP, CIDR or hostname); no packet can leave the machine for any other address. The restriction is enforced by the operating system firewall.
- Scripts **cannot write live tags**. Field writes only happen through separate approval cards.
- Timeout and output caps apply; the working folder is deleted after each run.
- **The code is shown to you and confirmed on every run.** The "apply without asking" mode and the always-allow list do not apply to this tool. Every run (code digest, exit code, duration) is written to the audit log and the compliance ledger.
- An **emergency stop** button kills all running scripts immediately and locks further runs; only the system administrator unlocks it.

The assistant can also **see your screen** when needed: for questions like "this looks broken" or "read this trend" it asks for permission; if you approve, a screenshot of the current screen is added to that conversation only.

New capabilities (Settings › Users › Permissions):

CAPABILITY	MEANING
Use the AI assistant	Chat with the assistant
MCP bridge	Connect with an external client (Claude Desktop, IDE) — separate from the assistant
Assistant may run scripts	Dangerous; never granted by a role default, only Root grants it individually
Assistant/MCP security administration	Script feature, approval mode, permanent allowances, external MCP servers — Root only

The assistant can also read your **license and package status**: how many tags you may still add, perpetual or subscription, when maintenance (the right to new versions) ends, whether a newer version exists and where to renew.