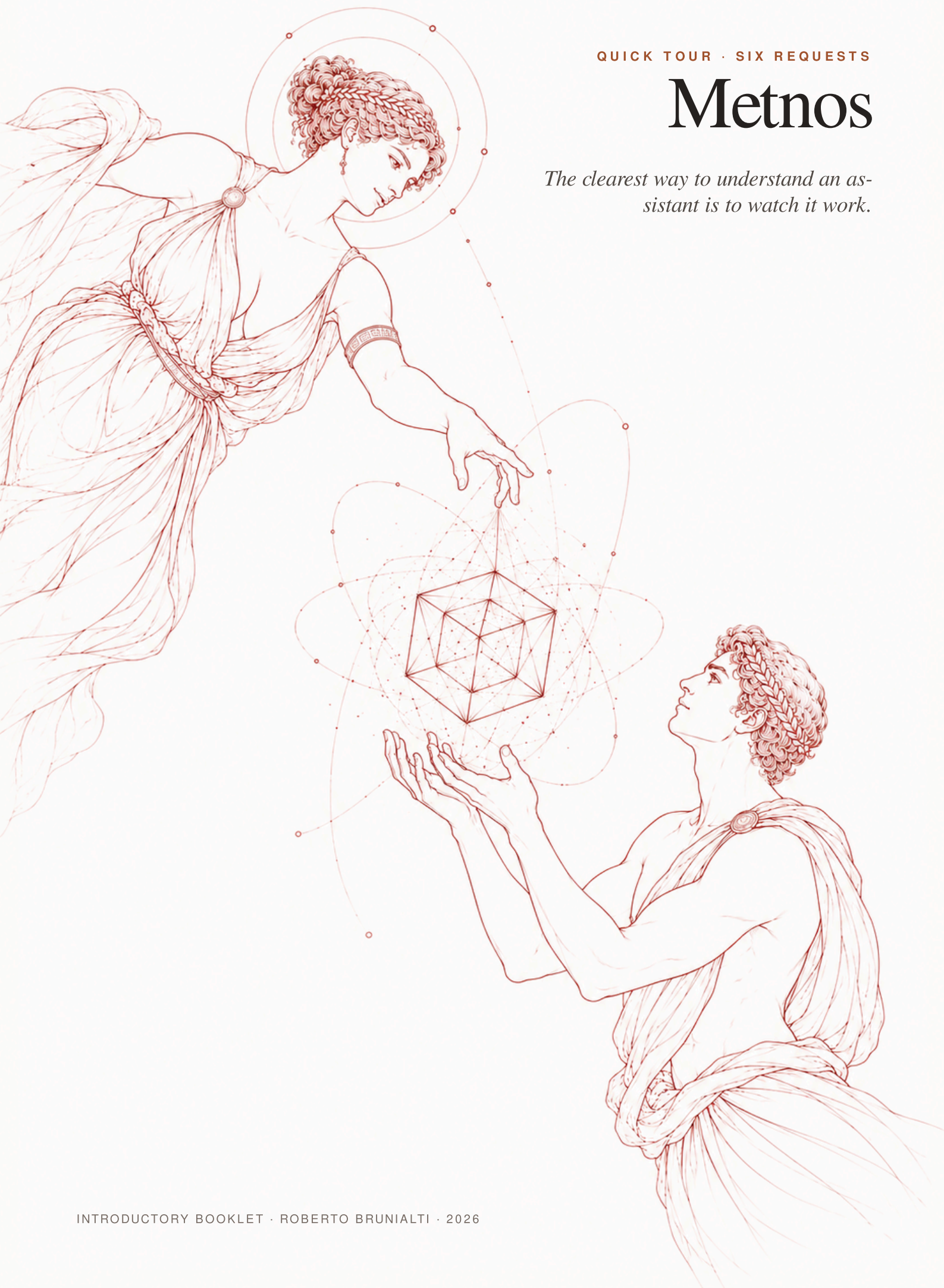


QUICK TOUR · SIX REQUESTS

# Metnos

*The clearest way to understand an assistant is to watch it work.*



A QUICK TOUR THROUGH EXAMPLES

# Metnos at work

There is no catalog of functions to memorize before you begin.  
These six conversations follow the whole path: install, ask, understand, act — and, when the operation truly supports it, undo.

*mētis (μῆτις), practical intelligence · noûs (νοῦς), mind*

|                                                  |                                           |
|--------------------------------------------------|-------------------------------------------|
| <div>4 commands</div> <div>TO BEGIN</div>        | <div>6 phases</div> <div>RESUMABLE</div>  |
| <div>14 pages</div> <div>OF ADMINISTRATION</div> | <div>Tutor</div> <div>INSIDE METNOS</div> |
| <div>undo</div> <div>ONLY WHEN REAL</div>        |                                           |

License: AGPL-3.0 · Code: GitHub · Installation: guide

**SIX REQUESTS AND ONE COMPARISON**

---

**01** From installation to the first answer

---

**02** Ask the Tutor for help

---

**03** Find files that are exactly identical

---

**04** From mail to a sheet, then consent

---

**05** Work on Windows and undo

---

**06** Follow a portal every morning

---

**07** Metnos, OpenClaw, and Hermes

# 1 · From installation to the first answer

Installation manages the whole instance, not just a Python package: an isolated environment, models, optional services, encrypted secrets, service units, the interface, and the Tutor catalog.

1

## STARTING OUT

### Four commands, then conversation

```
git clone https://github.com/brunialti/metnos.git
cd metnos
bash install/bootstrap.sh --check
bash install/bootstrap.sh
```

The preflight is read-only. The installer leads you through six phases, verifies each outcome, and records completed work. If a download or the Tutor build is interrupted, the same command resumes from the last valid point.

“ What time is it, and which time zone are you using?

#### ANSWER

It is 09:42 in Europe/Rome (UTC+02:00).

#### WHY THIS RESULT IS TRUSTWORTHY

An open HTTP port proves only that a process responds. This first request crosses classification, planning, an executor, and final presentation: it is a minimal proof of the complete chain.

**Installation:** guided, idempotent, and resumable · **Environment:** owned by the installation root, separate from user data · **First access:** one-use administrative link

## 2 · Ask the Tutor for help

You do not need to know the shape of Settings or guess a capability's technical name. Tutor answers questions about Metnos before the operational planner can mistake the question for an action.

2

### ORIENTATION

“Where can I see the configured embedder?”

“ Show me which embedder is configured under Settings > System > Models, and guide me there from the web chat.

### ANSWER

**Path in the web chat:** Settings → System → Models. The page shows the effective, editable configuration for language, embedding, and vision-language models. Open the official interface guide

`metnos.com/en/interface#settings`

.

If the question arrives from Telegram, Tutor says that the path must be opened in the web chat. It can also describe the page attested by the current instance, give its address, and say what is missing rather than inventing a menu.

### WHY THIS RESULT IS TRUSTWORTHY

Tutor is compiled locally from published documentation, admitted executor manifests, and permitted runtime registries. Changes to those sources rebuild and sign the catalog; internal notes and personal files are never Tutor sources.

**Settings:** 14 pages in four areas · **Validated languages:** Italian and English · **Boundary:** Tutor explains; it never acts quietly

## 3 · Find files that are exactly identical

A broad request should not yield a convenient but incomplete answer. Metnos keeps the limit on what fits in chat separate from the limit on the work that must be completed.

3

### COMPLETENESS

### Count everything, compare content, create the sheet

“Inspect the Pictures directory on the Metnos server: count every file and directory, then find every image that is certainly duplicated. Create a sheet with both paths and filenames for each pair.

#### EXECUTOR CHAIN

complete scan → counts → group by size → full SHA-256 → sheet containing every relation

#### ANSWER

In the illustrated run, the complete count found 34,060 files, 928 directories, and no symbolic links: 34,988 items in all. The duplicate search then examined the 32,239 image-type files, finding 2,809 exact-duplicate groups and 3,084 redundant copies. I created “Duplicate images” with 3,084 data rows.

#### WHY THIS RESULT IS TRUSTWORTHY

“Certainly duplicated” means identical content verified by the full digest — not a similar size or a visual probability. A row limit may keep the chat readable; it cannot silently reduce the data handed to the spreadsheet.

**Search:** recursive and parallel within the instance budget · **Outcome:** complete counts and declared truncation · **Sheet:** headings map semantically to fields, never by position

## 4 · From mail to a sheet, then consent

One sentence can join several domains. Typed results pass between steps; the model does not need to copy the data and cannot freely pick an account or provider.

4

### COMPOSITION

#### Search, extract, and deliver

“ Find June's invoice emails, extract date, supplier, invoice number, and amount, then create a sheet sorted by date.

#### EXECUTOR CHAIN

message search → structured extraction → sort → create sheet

#### ANSWER

I created “June invoices” with 27 data rows. Three messages had no recognizable amount and are marked incomplete.

“ Send the sheet to my accountant.

#### May I send a message with an attachment?

Recipient: accountant@example.com · attachment: June invoices.xlsx

External effect · cannot be undone after sending

#### WHY THIS RESULT IS TRUSTWORTHY

Reading and creating a local document do not require the same decision as sending it outside. Consent is bound to the user, recipient, attachment, and this proposal. A sent email is never described as undoable.

**Account:** selected by configuration and explicit context · **Secrets:** resolved only at execution time · **Errors:** incomplete rows are declared, never filled by plausibility

## 5 · Work on Windows and undo

The server retains the plan, policy, consent, and audit trail. A paired client carries only compatible execution to the device where the data lives.

5

### REMOTE DEVICE

## The photos remain on the PC

“ On OFFICE-PC, organize the photos in Pictures into subfolders by year without changing their names.

### ANSWER

Completed on OFFICE-PC: 486 photos moved, 12 subfolders created, 0 errors. Undo is available.

“ Undo the last operation.

### ANSWER

I restored 486 photos to their original paths on OFFICE-PC and removed the 12 folders that remained empty.

### WHY THIS RESULT IS TRUSTWORTHY

Undo is not a model-generated repair. The runtime retains the source/destination pairs and device identity, then executes the inverse on the same PC. If the PC is off, Metnos never falls back silently to the server: it waits or reports that the device is unavailable.

**Connection:** outbound client, signed invocations · **Authority:** paths granted for this call · **Undo:** idempotent and limited to recorded effects



## 6 · Follow a portal every morning

When the route may change but the purpose remains narrow, an executor may observe a page and choose among enumerated actions. That does not turn it into an unbounded agent.

6

### WEBSITE AND SCHEDULE

## From the school portal to Telegram

“ Every morning at 7, sign in to the school portal, collect homework, marks, and notices, read attached PDFs too, and send the five most important updates to my daughter on Telegram.

### EXECUTOR CHAIN

07:00 schedule → site session → encrypted credential login → pages and PDFs → summary → authorized user delivery

### ANSWER

#### School — today's summary

1. Notice about Friday's strike
2. Mathematics test postponed
3. New weekend homework
4. Parent meetings open for booking
5. Latin mark published

### WHY THIS RESULT IS TRUSTWORTHY

The mandate, allowed site, recipient identity, maximum item count, and output format remain fixed. If login expires, a second factor appears, or the site changes beyond what the executor can interpret, the task reports the block; it never invents a summary.

**Credentials:** encrypted and outside the prompt · **Schedule:** visible under Settings → Activity → Scheduler · **Users:** authorization and delivery remain independent for each identity

## 7 · Metnos, OpenClaw, and Hermes Agent

All three can be installed locally and accept natural-language requests. The decisive difference is where each one places management, guidance, authority, and recovery. This table compares behavior described by the projects' official sources; it is not an absolute score.

OpenClaw and Hermes both offer guided setup and capable operating surfaces. Metnos is distinguished by the depth of the integrated whole: one installer provisions and verifies the stack, one Settings area administers the instance, and Tutor answers from the capabilities and configuration that instance actually admits.

| AREA                     | OPENCLAW                                                                                                                            | HERMES AGENT                                                                                                               | METNOS                                                                                                                                                                              |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Everyday use</b>      | A personal assistant across WebChat and many messaging channels.                                                                    | CLI, desktop, and several channels, with automation and delegation.                                                        | Web chat and Telegram turn an ordinary-language outcome into a typed workflow across server, providers, and devices.                                                                |
| <b>Installation</b>      | One-line installers and guided onboarding on several operating systems.                                                             | Desktop packages or a one-line installer, a setup wizard, and native Windows support.                                      | A read-only preflight and six resumable phases configure environment, models, sidecars, secrets, services, and Tutor, then verify the complete instance.                            |
| <b>Management</b>        | Control UI and CLI for gateway, channels, agents, and configuration.                                                                | Desktop, TUI, CLI, and dashboard provide operational surfaces.                                                             | Settings unifies 14 authenticated pages for activity, executors, memory, editable models, services, safety, users, and devices.                                                     |
| <b>Built-in guidance</b> | Documentation, setup flows, and configuration commands are available from chat and the UI.                                          | /help, setup, doctor, and extensive documentation guide the operator.                                                      | The semantic Tutor is compiled from installed sources and manifests: it answers in the user's language, observes permitted registries, and provides exact paths and official links. |
| <b>Security boundary</b> | The main personal session runs on the host by default; sandbox modes, tool policies, pairing, and security audits are configurable. | Dangerous-command approval, protected paths, pairing, and optional multi-layer isolation provide defense in depth.         | The planner does not receive a general shell: it sees admitted, signed executors. Schema, per-call authority, isolation, policy, consent, and post-conditions govern every path.    |
| <b>Undo and recovery</b> | Recovery is session- or tool-specific; the core documentation does not define a uniform cross-domain inverse-operation contract.    | /undo manages conversation state; opt-in checkpoints restore project files changed by file tools and destructive commands. | Supporting executors declare the inverse and the runtime records resources, account, and device. Irreversible effects are stated explicitly.                                        |
| <b>Extension</b>         | Skills and plugins teach new procedures and add tools.                                                                              | Installed or learned skills, MCP, and toolsets extend the agent.                                                           | Code becomes routable only after its name, schema, authority, signature, tests, and admission state satisfy the shared contract.                                                    |

**The recovery models are not equivalent.** Hermes checkpoints can restore everything below a tracked project root, even when no individual tool declared an inverse; inside that tree, Metnos has no equivalent snapshot. Metnos inverse operations instead reach effects a project copy cannot touch — for example a message moved on an IMAP server or files moved on a paired PC — but only when the responsible executor declares and records the reverse. Neither system can recall an email that has already been sent.

Comparison checked on 31 July 2026 against the official sources: OpenClaw

`github.com/openclaw/openclaw`

, OpenClaw security

`docs.openclaw.ai/cli/security`

, Hermes Agent

`github.com/NousResearch/hermes-agent`

, and Hermes checkpoints

`hermes-agent.nousresearch.com/docs/user-guide/checkpoints-and-rollback/`

. These projects evolve; this is a dated view of documented behavior, not an eternal ranking.

# Continue from here

---

## Try more requests

Operational domains with examples ready to use.

[metnos.com/en/domains](https://metnos.com/en/domains)

## Explore the interface

The 14 Settings pages and their exact paths.

[metnos.com/en/interface](https://metnos.com/en/interface)

## Understand the architecture

Contracts, authority, Tutor, memory, and devices.

[metnos.com/en/architecture/](https://metnos.com/en/architecture/)

Metnos · a quick tour through examples · AGPL-3.0 code · [metnos@metnos.com](mailto:metnos@metnos.com)