HACOY

How a small fashion house runs on local AI

How a fashion house in Munich runs its day on AI models that live on its own machines, what we measured, and what did not work.

130	224	24,124	99.8 %
agent roles	scheduled jobs	model calls in 30 days	of them succeeded, all in house

HACOY Research Note 01 · Munich · October 2026

Every figure comes from our own records, read on 18 October 2026, and is named in the sources at the end. Nothing in this note names another brand.

In short

HACOY is a fashion house with one founder and a small team. Behind it runs a system of 130 AI agent roles and 224 scheduled jobs that read the mail, draft replies, check product facts, research questions and keep the records in order. All of it runs on six Apple computers that we own, in Munich. In the last 30 days the system made 24,124 calls to a language model. Every call in our log was answered by a model on our own machines, and 99.8 percent of them succeeded.

This note describes how the system is built, what we measured, and what did not work. We write it down because we learned most of it the slow way, and because a small house is a good place to see how this works in practice.

Why local

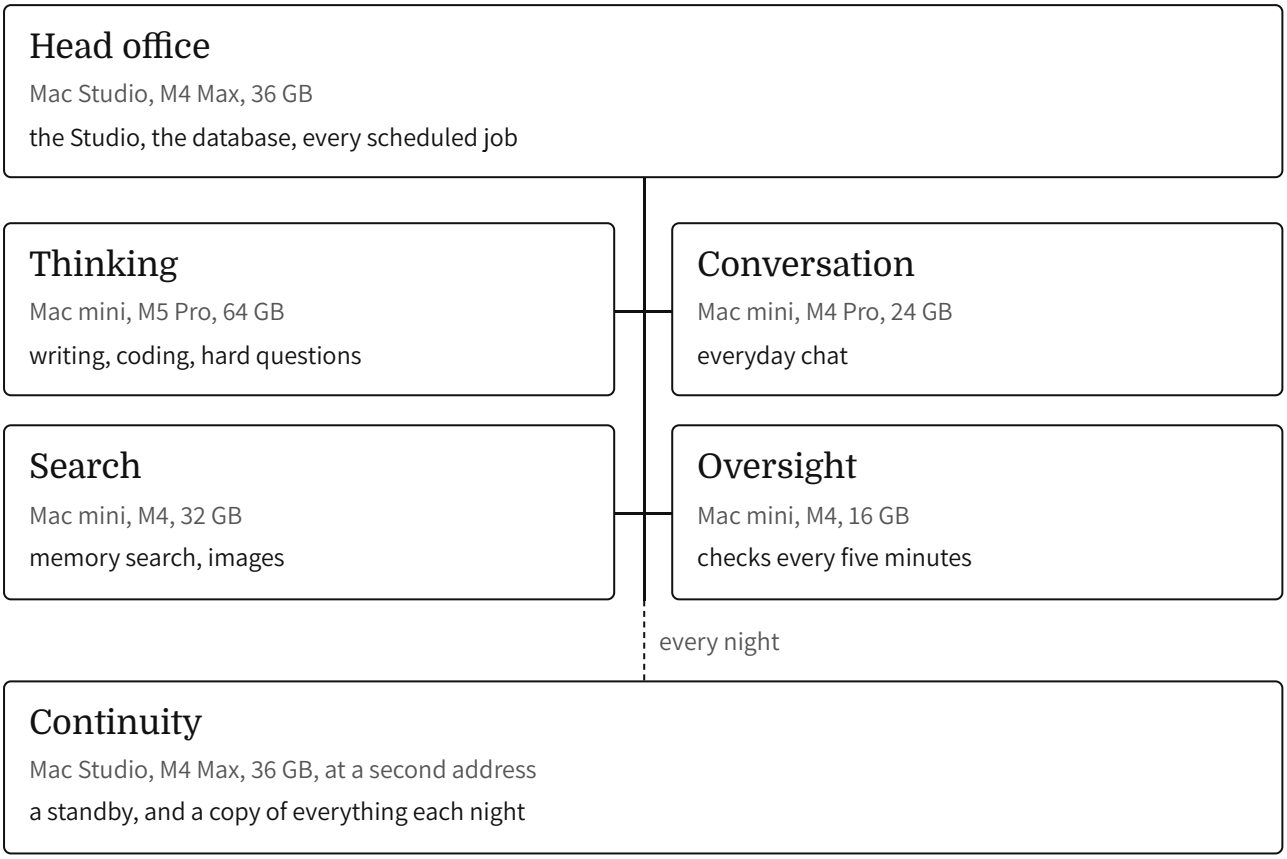
We decided early that the agents which run on their own, around the clock, would use only models on our own machines. There are three reasons.

The first is privacy. Customer mail, orders and our own notes never leave the house to be read by someone else's model. The second is cost that does not grow with use: a loop that runs every five minutes costs the same electricity whether it is useful or not, and there is no bill per word. The third is learning. Every prompt that fails on a small model teaches us something about the task, and the answer is usually to make the task smaller, not the model larger.

We build and repair the system itself with a cloud coding assistant, with a person reading every change. What runs the company day to day is local.

The machines

Six machines, planned like a team



Source: our machine register and role plan, 23 and 24 September 2026. Each machine has one job, a deputy and a target.

Role	Machine	Memory	What it does
Head office	Mac Studio, M4 Max	36 GB	the Studio app, the database, every scheduled job, the one input field
Thinking	Mac mini, M5 Pro	64 GB	the writing model, the coding model, the model for hard questions
Conversation	Mac mini, M4 Pro	24 GB	everyday chat in the background
Search	Mac mini, M4	32 GB	the embeddings that let the system find things in its own memory, and reading images
Oversight	Mac mini, M4	16 GB	checks every five minutes that nothing has quietly stopped
Continuity	Mac Studio, M4 Max	36 GB	a standby at a second address, a nightly copy of everything

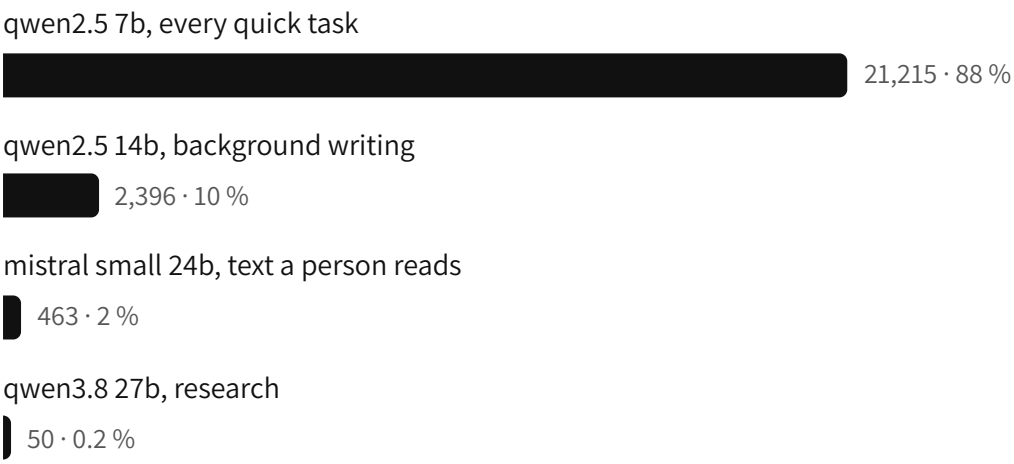
We planned the machines the way one plans a team. Each one has a single job, a list of what it never does, a deputy who covers when it is out, and a target that can be checked. The head office machine is the only one with many duties, which is why it has a twin.

The models

Model	Size on disk	Its job	Calls, last 30 days
qwen2.5 7b	4.7 GB	every quick task: sorting, short drafts, persona chat	21,215
qwen2.5 14b	9.0 GB	background writing that can take a few seconds more	2,396
mistral small 24b	14.3 GB	writing that a person will read	463
qwen3.8 27b	30 GB	research and questions that need thinking first	50
qwen3 coder 30b	18.6 GB	the coding assistant	
qwen2.5 vl 7b	6.0 GB	reading images	
nomic embed, bge m3	0.3 and 1.2 GB	turning text into vectors for search	

Nine in ten calls go to the smallest model. That was not the plan at the start; it is what the measurements taught us.

Where the 24,124 model calls of the last 30 days went



Source: the Studio's model log, the 30 days to 17 October 2026. Every call answered on our own machines; 99.8 percent succeeded.

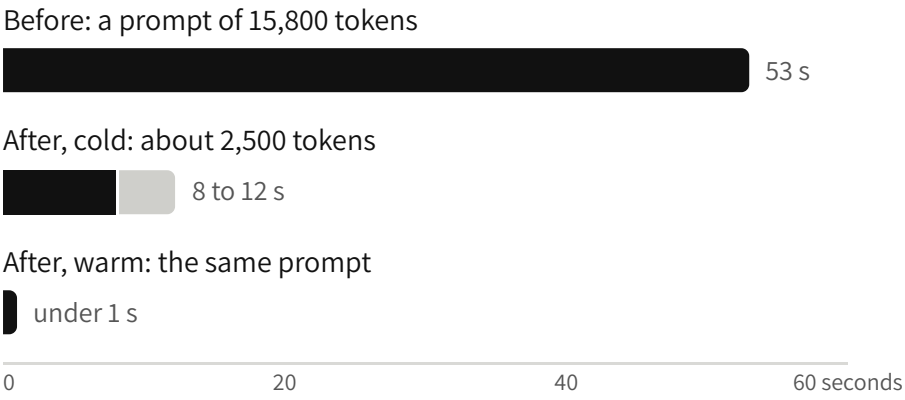
What we measured

A small model kept warm beats a larger one kept cold. On a Mac mini the 7b model answers in 3.1 seconds from cold and 0.44 seconds when it is already loaded. The 14b model needs about 60 seconds to load again after it has been pushed out of memory, which made it unusable for chat on a 36 GB machine. On 25 June 2026 we moved every chat role to the 7b and kept it loaded.

The largest model that fits is not the largest that works. A 32b model on the 36 GB head office machine ran out of memory and returned empty answers. We retired it from every path on 17 June 2026. About 12 GB of the 36 are free for models once the rest of the system is running, so the head office machine holds one model at a time.

The prompt is the slow part. Our in house assistant once started each answer with a 15,800 token persona and handbook. On the 64 GB machine that meant 53 seconds of reading before the first word. Cut to about 2,500 tokens, a warm answer starts in under a second and a cold one in 8 to 12 seconds.

Seconds before the first word of an answer

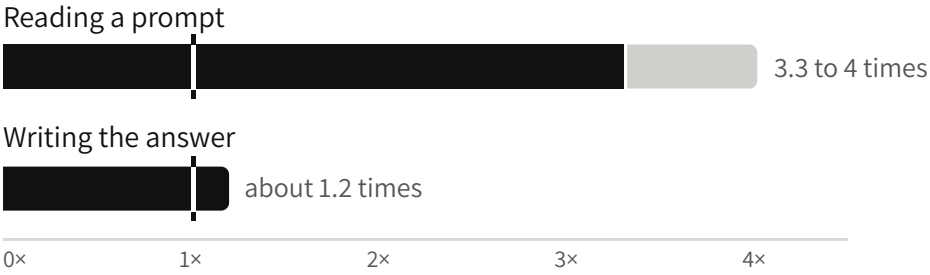


Source: our assistant on the 64 GB Mac mini, timed on 4 October 2026. The grey part of a bar is the range.

Speed on the 64 GB machine, measured on 4 October 2026: the writing model starts in about 2 seconds and writes about 20 tokens a second, the coding model about 90 tokens a second, the research model about 10 tokens a second after it has thought.

More memory buys room, not speed of writing. Apple's own figures for the M5 generation show prompt reading 3.3 to 4 times faster than on the M4, but writing only about 1.2 times faster, because writing is bound by memory bandwidth. A bigger machine lets us run bigger models and longer prompts. It does not make the same model write much faster.

How much faster the M5 generation is than the M4



The upright black line is the speed of the M4. Source: Apple Newsroom, 25 August 2026. Writing is bound by memory bandwidth, so a bigger machine buys room more than writing speed.

How an agent is built here

Every agent starts as one sentence from the founder, for example "a research department, for HACOY and for myself". It then becomes a role with a name and a short prompt, a context builder that pulls only the records this role needs, a short kit of house rules loaded at the top, a loop that runs on a schedule, and a queue where its work waits for a person.

The kits matter more than the models. Each kit is one short file per topic, with one dated line per rule. The voice kit, for example, says that HACOY writes warm and precise, never loud, and checks every text with a small validator before it can leave. A small model follows a short, specific file far better than a long handbook.

The founder has one input field. Whatever he types, the system files, drafts or researches, and returns one line. Nothing reaches a customer without his approval, unless a named switch for that one kind of message has been turned on after review.

What did not work

Rules that are only read change nothing. An audit on 30 August 2026 found that our house rules had been served to sessions 425 times and had changed an outcome zero times. Rules that can be checked by a machine became a blocking validator instead. A text that breaks one does not leave.

A queue nobody can read is not oversight. By 13 October 2026 about 19,700 items were waiting for approval. One writing loop had produced 14,364 drafts and not one was kept. One in 102 drafted LinkedIn posts was approved. Since 15 October a nightly check measures how often each producer's work is kept and flags any that falls below one in ten, so it can be paused. What is old is archived, never deleted.

Agents need a register. Of 217 scheduled jobs on 13 October, only 36 could be traced to a named agent role. Some loops ran without producing anything. A count of agents measures effort; what we now count is what each one delivered and whether a person kept it.

What we take from it

A small house does not need a large model to run on AI. It needs small tasks, short rules, a person at every message that leaves, and measurements it actually reads. The machines cost once; the learning compounds.

We will publish further notes as we measure more. Students and researchers who would like to work with us on questions like these are welcome on our research page.

Sources

The figures come from our own records: the Studio's model log for the 30 days to 17 October 2026, the machine register and role plan of 23 and 24 September 2026, the agent stack map of 22 September 2026, the assistant timings of 4 October 2026, and the Studio audit of 13 October 2026, and the producer check of 15 October 2026. Apple's figures for the M5 generation are from the Apple Newsroom, 25 August 2026.