

Private AI: when running models on local hardware makes sense

Local AI can be a useful architecture choice. The case for it should begin with a business requirement, not a hardware purchase.

IN THIS GUIDE

First, separate private from local

When local hardware deserves consideration

When hosted AI may be the better choice

Buy for the workload, not the model's headline size

Compare total cost, not just usage fees

A practical pilot checklist

“We want AI, but we don't want our information leaving the business.”

That is a reasonable starting point. It is not yet an architecture.

Before buying a powerful workstation, decide what the system needs to do, which information it will handle, and what “private” means for your organization. A useful local AI system is a business workflow with clear boundaries—not simply a model running under someone's desk.

First, separate private from local

For this guide, **local AI** means running the model on hardware you control, such as a workstation or an on-premises server. **Private AI** is the broader goal: controlling access, permitted uses, retention, and where information is processed. A dedicated hosted environment may also meet those requirements, depending on the deployment and contract.

Do not infer privacy from an application's location alone. Ollama, for example, distinguishes local execution from cloud features and documents a local-only mode.^[4] Check the entire application, not just the model runner.

Ask whether document processing, search, embeddings—the representations used to retrieve relevant text—error reporting, or backups depend on external services. A local model connected to a cloud document processor is not an entirely local workflow.

When local hardware deserves consideration

You have a specific data-boundary requirement. Perhaps a contract or internal policy restricts where confidential documents may be processed. Identify that requirement in writing, then test whether the proposed design actually satisfies it.

The task is narrow enough to evaluate. Consider a hypothetical manufacturer extracting fields from a consistent set of internal service reports. That is a more concrete pilot than promising a general-purpose assistant that can answer anything about the business.

You need operation without a reliable internet connection. Validate the whole workflow offline, including authentication and document access. A model running locally is not enough if another required component becomes unavailable.

Demand is predictable and someone can operate the service. Regular workloads may justify dedicated capacity. But the business also needs an owner for patching, monitoring, access control, recovery, and model updates.

Treat these as reasons to investigate—not proof that local deployment will be cheaper or better.

When hosted AI may be the better choice

Favor a hosted pilot when your use case is still changing, demand is intermittent, or the models you can run locally do not meet your quality requirements.

Also consider hosted options when nobody is responsible for maintaining another production system. Transferring infrastructure work to a provider does not remove your responsibility for data handling, but it changes the operational burden.

For healthcare, local hardware is not a shortcut around HIPAA. HHS describes the Security Rule as technology-neutral; its safeguards apply to regulated entities' ePHI regardless of the chosen technology.^[3] Conversely, HIPAA does not categorically prohibit cloud processing when the applicable agreements and other requirements are met.^[1]

Buy for the workload, not the model's headline size

Before selecting hardware, test representative documents, expected response lengths, and realistic simultaneous use. Ollama's documentation notes that available memory affects model loading and that parallel requests and context length affect memory requirements.^[4]

Evaluate:

Accuracy:	Does the output meet the task's acceptance criteria?
Review burden:	How much correction does a person need to make?
Responsiveness:	Is the wait acceptable during actual work?
Capacity:	What happens when several people submit requests together?
Recovery:	Can the business keep working if the machine fails?

Check the model's license and permitted commercial uses separately from the software used to run it. Being downloadable is not a sufficient purchasing criterion.

Compare total cost, not just usage fees

For local deployment, include hardware, installation, electricity, support time, backups, spare capacity, security work, and eventual replacement.

For hosted deployment, include usage, storage, integrations, monitoring, support, and review time. Compare both options at similar quality and availability targets. An inexpensive answer is not inexpensive if someone must repeatedly repair it.

Avoid a universal break-even promise. Price the workload you actually expect, including quiet periods and peak demand.

A practical pilot checklist

- ☐ Choose one bounded task and a representative evaluation set.
- ☐ Document what may leave the environment and what must remain inside it.
- ☐ Compare a local option with an approved hosted alternative.
- ☐ Measure quality, review effort, response time, and operating cost.
- ☐ Test access restrictions, outbound connections, and recovery.
- ☐ Assign an operator and agree on acceptance criteria before purchasing production hardware.

A hybrid design can also be sensible: keep restricted work local and route only explicitly approved information to hosted services. Do not let the system silently fall back to the cloud when a local request fails.

Bottom line: Local AI makes sense when control, workload fit, and operational readiness justify it. Hardware should follow that decision—not substitute for it.

Considering private AI? Start with a workload and data-flow assessment before choosing a model or buying a server.

Sources

[1] <https://www.hhs.gov/hipaa/for-professionals/special-topics/health-information-technology/cloud-computing/index.html>

[3] <https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/index.html>

[4] <https://docs.ollama.com/faq>