Local AI for Cybersecurity

Uses, effects, risks and hardware value

ViWeb Technology | viweb.net Research edition 1.0 | 17 November 2026 For technicians and AI users | US purchasing baseline, USD

Executive Summary

Local AI is a practical option for assisting with sensitive logs, source code and internal documentation when the organization can secure the complete application. Its strongest starting point is a supervised assistant: it explains evidence and proposes work while a technician verifies the conclusion and controls consequential actions. Keeping inference on a local device changes where data is processed; it does not make the model trustworthy or the host secure. [1] [2]

Use it first for bounded work. Start with incident summaries, questions over approved runbooks, configuration review and draft detection queries. Preserve the original evidence, require source references and test output before use. These are recommended pilot uses, not measured ViWeb outcomes.

Buy for the workload that passes evaluation. A compact Mac is a reasonable candidate for a single-user assistant. Ryzen AI Max+ 395 systems merit a comparison when larger models need 64GB or 128GB of shared memory. A discrete-GPU PC merits a comparison when the selected software needs CUDA. Memory capacity, token speed and useful security outcomes are different measures; this report does not establish one universal price-to-performance winner.

Keep authorization outside the model. Prompt injection can arrive in the material a technician is investigating. Retrieval controls, limited tools, validated outputs and host isolation should constrain what a successful attack can do. A prompt telling an assistant to be careful is insufficient. [2]

Reading map

Start here	What it answers
Pages 2-6	Useful tasks, evidence of effects, risks and a controlled deployment
Pages 7-11	Memory sizing, US hardware prices, benchmark limits and operating cost
Pages 12-15	Purchase tests, technician workflows, adoption gates and evidence limits
References	Clickable primary sources and dated purchasing notes

Desk research with AI assistance. No devices were purchased or benchmarked for this report. Prices are dated evidence, not offers. Publication does not imply a new ViWeb service commitment.

01 / What local AI can do for a technician

Define the boundary before choosing the model

In this report, **local AI** means inference on a device controlled by the user or organization. A local desktop, an internal LAN server and an isolated workstation have different access and failure boundaries. **Offline** additionally requires the workflow to operate without external connections; **air-gapped** requires an enforced separation and a controlled transfer process. A product having a local interface proves none of these conditions.

The scope is mainly generative language models, retrieval and tool-assisted workflows. Traditional statistical detection, antivirus engines and large-scale security analytics remain useful, but their effectiveness cannot be inferred from an LLM comparison. Training a foundation model is outside the proposed desktop use case.

Recommended pilot tasks

Task	Useful output	Required check
Web, authentication or firewall logs	Timeline, grouped events, hypotheses	Reconcile with raw records, timezones and deterministic queries
Configuration and source-code review	Explanations, suspicious patterns, proposed fixes	Inspect the exact code and reproduce the issue in a test environment
Threat-intelligence reading	Summary linked to advisories	Confirm affected product, version, prerequisites and advisory date
Detection engineering	Draft SQL, Sigma or other query logic	Validate schema, syntax, false positives and known malicious cases
Internal runbook search	Answer with document references	Enforce access before retrieval; check currency and actual passages
Incident and recovery reporting	Evidence-based draft for the technician	Verify chronology, scope and every claim of compromise or recovery

These examples are proposed workflows. Their value depends on the selected model, input quality and the time spent correcting its output. An attractive explanation is not confirmation of a vulnerability or incident.

Start with retrieval before fine-tuning

Retrieval-augmented generation (RAG) supplies selected documents at inference time. It can make answers easier to trace and keep operational knowledge current, but permissions must follow the source documents into the index. Fine-tuning changes model behavior and adds a separate data and model maintenance burden. Neither technique removes prompt injection. [3] [2]

For a first pilot, prefer a small, approved knowledge collection and read-only access over a broad index of every customer and server.

02 / Effects: useful assistance is task dependent

Productivity gains require a comparable baseline

A Microsoft-authored phishing-triage preprint recruited 167 professional analysts and excluded 13 responses. It reports up to 6.5 times as many true positives per analyst minute with its purpose-built agent. Participants used curated queues, including simulated agent mistakes; queue ordering contributed substantially to the result. This supports testing a specific assisted workflow. It does not establish the same improvement for an open-weight model running locally. [4]

A useful counterexample is METR's early-2025 coding trial: 16 experienced developers working on 246 real tasks took 19% longer when AI tools were allowed. That result concerns a historical toolset and particular repositories. Its February 2026 follow-up describes selection effects that complicate later estimates. These studies should not be averaged or treated as a current estimate for all technicians. [5] [6]

Practical implication: measure end-to-end completion time and correctness on your own tasks, including waiting, verification and rework. A faster first draft can still increase the cost of a finished investigation.

Local deployment changes the operating tradeoff

Potential benefit	Cost or condition that accompanies it
Less routine transfer of confidential prompts to a hosted model	The client, embeddings, logs, backups and tools must also follow the intended data boundary
Continued operation during an internet outage	Models and dependencies must already be available and the complete workflow tested offline
Control over versions and update timing	The organization becomes responsible for patching, regression tests and rollback
Predictable ownership cost at sustained use	Idle capacity, administration, power and review time can dominate low-volume use
A shared internal assistant	Authentication, customer isolation, scheduling and availability become service responsibilities

These are architectural opportunities and tradeoffs, not measured benefits of a specific device.

Attackers benefit from AI as well

The NCSC's 2026 assessment expects AI to increase the scale, speed and efficiency of existing cyber operations, with uneven uptake and continued human direction. It also highlights shadow-AI exposure. This is a threat assessment, not a count of attacks caused by local models. [7]

Local open-weight tools can reduce dependence on hosted providers and their usage controls. That is a dual-use property, not evidence that local AI uniquely causes cybercrime. Defenders still need patching, strong authentication, tested recovery and reliable evidence collection.

03 / Risks that remain inside the local boundary

Treat the assistant as an untrusted interpreter

The table maps published risk categories to technician scenarios. Scenarios and controls are recommendations for a pilot; they are not claims that ViWeb or a named product has suffered these failures. OWASP references below identify the reviewed 2025 entries.

Risk	Technician scenario	Control to test
Indirect prompt injection [2]	An attacker-controlled log field or support document redirects the assistant	Treat content as evidence; limit resulting capabilities and test malicious inputs
Excessive agency [8]	An assistant can alter accounts, block traffic or delete files beyond its task	Narrow tool APIs; authorize every action in ordinary code; require review of impactful changes
Unsafe output handling [9]	Generated shell, SQL or HTML reaches an execution surface	Validate types and paths; parameterize queries; encode output; test proposed commands separately
Retrieval leakage or poisoning [3]	One customer's query retrieves another customer's records or a poisoned runbook	Filter by identity and customer before retrieval; enforce ingestion and deletion controls
Unsafe model loading [10]	A downloaded package executes code during initialization	Prefer data-only formats where supported; pin revisions; inspect required custom code
Supply-chain compromise [11]	A model adapter, runtime package or tool update changes trusted behavior	Record provenance and dependencies; stage updates; preserve a rollback route
Resource exhaustion [12]	Long requests or repeated tool loops occupy the only analysis workstation	Cap context, output, concurrency, queue depth, runtime and total tool actions
Confabulation and overreliance [1]	The model invents an indicator, advisory, diagnosis or claim of successful cleanup	Require verifiable evidence; score factual errors; permit an explicit uncertain answer

Offline does not remove the attacker from the data

A disconnected machine can still receive a malicious file, poisoned document or compromised package through a transfer process. Restricting network egress can reduce exfiltration paths, but cannot make a wrong answer correct or stop destructive operations that the assistant is allowed to perform locally.

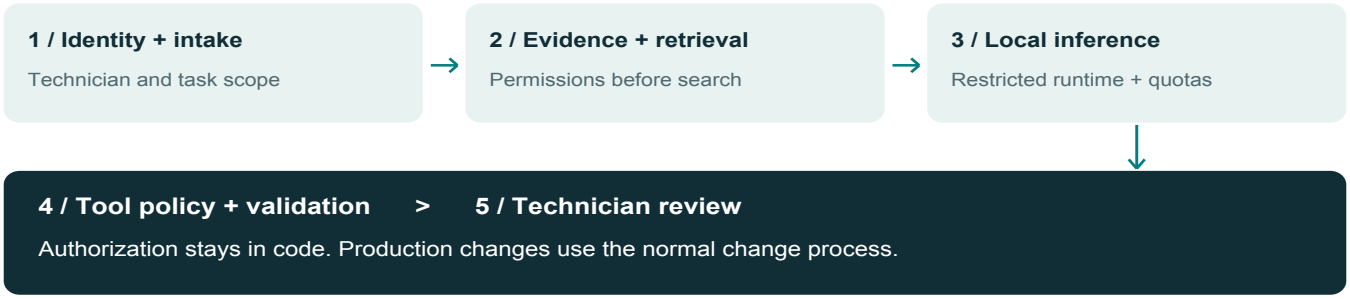
Sensitive material can also persist in chat history, temporary files, crash dumps, indexes and ordinary backups. Protect these as operational data, with explicit retention and access rules. The deployment must include the user interface and its supporting stores, not just model weights. [13]

A model refusing a harmful request is useful behavior. It is insufficient as the sole boundary protecting credentials, customer data or production tools.

04 / A reference design with enforceable boundaries

Separate evidence, inference and action

The following is a proposed deployment pattern for a small technical team. It is a design recommendation, not a representation of ViWeb's current infrastructure.



1. Intake: authenticate the technician, classify the task and select the customer or project scope. Remove secrets that are unnecessary to answer the question.

2. Evidence: ingest approved files through a restricted parser. Preserve originals, source identifiers and timestamps. Apply document permissions before retrieval.

3. Inference: run the approved model and local embedding service under a dedicated account. Bound memory, requests and runtime. Return answer passages with evidence identifiers.

4. Action: validate any proposed tool call against an allowlist, parameter schema and the user's permissions. Keep broad shells and production credentials outside the first pilot.

5. Review: show the exact proposed operation and target to the technician. Use the normal change process for consequential actions; retain the decision and result.

This pattern applies ordinary isolation, access control and monitoring principles to an AI workflow. It should reduce the consequences of a model mistake without assuming that the model can reliably police itself. [13] [8]

Data handling rules worth making explicit

Data	Pilot rule
Credentials, session tokens, private keys	Exclude from prompts and retrieval; tools obtain narrowly scoped credentials outside the model
Customer logs and source code	Use only within the agreed purpose and customer boundary
Retrieval indexes and embeddings	Inherit source sensitivity; implement access, retention and deletion propagation
Model and tool audit records	Record versions, action metadata and evidence IDs; minimize raw sensitive content
Model outputs and reports	Apply the same classification as the evidence they contain

05 / Set up a small assistant safely

Choose the runtime by model and platform support

llama.cpp is a useful cross-platform inference baseline with several accelerator backends. **MLX** is an Apple-silicon option. **Ollama** simplifies local model operation and exposes a local API. The exact model architecture, quantization, backend and tool-calling behavior still require validation. Installation success does not establish GPU acceleration or feature parity. [14] [15] [16]

For Ryzen AI Max systems, inspect the current AMD compatibility guidance for the intended OS, kernel, driver and framework. AMD's RDNA3.5 documentation describes memory setup and kernel requirements. A successful Vulkan inference test does not automatically qualify a ROCm training or serving workload. [17]

Example: verify an intended local-only Ollama workflow

The current documentation provides the following local-only setting. Apply it through the environment of the actual server process, restart the server and verify the resulting behavior for the installed version. [16]

```
OLLAMA_NO_CLOUD=1
```

Alternatively, the documented server configuration contains:

```
{
  "disable_ollama_cloud": true
}
```

Ollama documents a default listener on `127.0.0.1:11434`. Keep a single-user pilot on loopback. Sharing it over a LAN requires an authenticated service boundary and network rules appropriate to the users. Cloud-feature settings do not control every connection made by a separate UI, plugin or operating system.

Acceptance checks before using confidential input

• Confirm the exact local model is selected and no cloud fallback, external embedding provider or web-search tool is active.

• Check outbound traffic while exercising the complete UI and workflow. Test inference after disconnecting external connectivity, with required assets already downloaded.

• Inspect the listening address, account permissions and access to customer folders. Try an unauthorized request from a second user or device.

• Confirm actual accelerator residency and memory consumption. In Ollama, inspect `ollama ps`; monitor the process as well.

• Inspect history, debug logs and retrieval stores for unwanted copies of test secrets. Confirm deletion and a usable retention policy.

• Test a staged update and rollback. Record model revision, file hash, runtime, driver, configuration and evaluation outcome.

For malware analysis, pass extracted text or metadata from an appropriate isolated analysis environment. Do not execute an unknown binary on an ordinary AI workstation simply to obtain a better explanation.

06 / Size memory before buying compute

Weight size is a floor, not the working set

A useful first estimate is **weight bytes = parameter count x bits per weight / 8**. The table uses decimal GB and idealized dense weights. Actual quantized files include scales, metadata and mixed-precision tensors. Runtime allocations, the key/value (KV) cache, the OS and other applications add memory.

Nominal model	Ideal 4-bit weights	Ideal 8-bit weights	Ideal 16-bit weights
8 billion parameters	4GB	8GB	16GB
14 billion	7GB	14GB	28GB
32 billion	16GB	32GB	64GB
70 billion	35GB	70GB	140GB
120 billion	60GB	120GB	240GB

Illustrative planning allowance: a nominal 32B 4-bit model might have a 20GB file. With 4GB of runtime buffers, 8GB of KV cache and 8GB for the OS and other services, the total is 40GB. These are assumptions, not measurements or a universal allocation. Measure the real model at the intended context and concurrency before selecting a 48GB or 64GB configuration.

Context and concurrency can change the answer

For a conventional transformer cache, an approximate KV allocation is:

```
2 x layers x KV heads x head dimension
  x bytes per cache element x cached tokens x sequences
```

For 32 layers, 8 KV heads, a head dimension of 128 and a 16-bit cache, 32,768 cached tokens occupy about 4GiB for one sequence; four such sequences occupy about 16GiB. Cache sharing, compression and other model architectures can change this. GB and GiB are different units.

A mixture-of-experts model may activate only some parameters per token while still needing the full collection of expert weights resident or managed by an offloading strategy. A low active-parameter count does not imply a comparably small download or working set.

Translate fit into a practical purchase

Start with roughly 24-32GB shared memory for a small single-user assistant, evaluate 48-64GB when larger models or contexts justify it, and compare 128GB systems when measured needs exceed the smaller tier. These are starting points for testing, not model-fit guarantees. CPU offload or swap can make a model start while missing the latency target.

07 / US hardware shortlist and dated prices

Compare the exact configuration and price type

Checked 17 November 2026. USD, before sales tax. Shipping, peripherals, support and additional licenses are excluded unless stated. Apple prices below come from its September 2026 **Consumer (MSRP)** column; other amounts are displayed vendor listings. No checkout or delivery promise was verified.

Candidate	Memory / storage	US price evidence	Value proposition to test
Mac mini M6	24GB / 512GB	$1,299 MSRP [18]	Compact single-user assistant with small models
Mac mini M5 Pro	24GB / 512GB	$1,699 MSRP [18]	Higher-bandwidth Mac; same capacity as the row above
Framework Desktop DIY, Max+ 395	64GB / SSD extra	$1,959 system price [19]	Larger-memory Linux or Windows candidate; price the full build
GMKtec EVO-X2, Max+ 395	64GB / 1TB	$2,199.99 listing [20]	Complete storage-equipped candidate in the 64GB tier
Mac Studio M5 Max	36GB / 512GB	$2,499 MSRP [18]	Evaluate when the workload needs its platform capabilities; capacity alone does not justify the step
Framework Desktop DIY, Max+ 395	128GB / SSD extra	$3,449 system price [19]	Higher capacity; add storage, OS and assembly requirements
MINISFORUM MS-S1 MAX, Max+ 395	128GB / 2TB	$3,959 displayed; quote required [21]	Network/expansion option; sale-ended message makes price uncertain
Beelink GTR9 Pro, Max+ 395	128GB / 2TB	$4,199 displayed sale price [22]	Complete higher-capacity candidate; confirm final price and support
GMKtec EVO-X5 Pro, Max+ PRO 495	192GB / 2TB	$6,799 listing [23]	Capacity-driven specialist option; workload must justify the premium

Do not compare a headline price with a different memory tier

The EVO-X2 page's $2,199.99 selection is **64GB/1TB**, while 128GB options are also shown. The MINISFORUM page displays promotional amounts alongside a sale-ended message. Those limitations can change a purchase decision and should survive any quotation or website excerpt.

Framework's configurator lists a 1TB SSD addition at $215. Adding that to the 64GB system gives **$2,174 before other options**, versus the EVO-X2's $2,199.99 listing. The $25.99 difference is too small to decide between them without confirming the complete build, OS, support, availability and workload results. This is component arithmetic, not an executed order. [19] [20]

Current Apple pages list M6 and M5 Pro Mac mini configurations. Older M4 results later in this report are historical measurements and must not be relabeled as M6 or M5 Pro performance. [24]

08 / Which platform deserves a closer look?

Mac mini: a compact candidate for bounded work

Apple lists 153GB/s memory bandwidth for the base M6 memory configuration and 170GB/s with 32GB, versus 307GB/s for M5 Pro. The M6 supports up to 32GB and M5 Pro up to 64GB. Shared memory also serves the OS and other applications; it is not all available for weights. Request a configuration-specific quote for 48GB or 64GB instead of applying the 24GB price. [24]

Recommendation: begin with a Mac mini when macOS supports the intended tools and the selected model fits with sufficient headroom. Test the more expensive configuration only if measured latency, memory or other useful work justifies it. Do not infer CUDA compatibility from the presence of an Apple GPU.

Ryzen AI Max+ 395: compare usable memory and complete builds

AMD lists a 256-bit LPDDR5x-8000 interface, up to 128GB of memory and Radeon 8060S graphics. The interface implies a theoretical 256GB/s bandwidth; that is a specification-derived ceiling, not a measured application rate. AMD describes up to 96GB of graphics allocation with 128GB systems. OEM firmware, OS and runtime determine practical allocation and usable capacity. [25] [26]

Recommendation: shortlist a 64GB system for mid-sized workloads and a 128GB system when the measured working set exceeds that tier. Confirm sustained cooling, accelerator use, firmware updates, storage replacement and a workable support route. Soldered LPDDR5x capacity makes an undersized purchase difficult to correct.

Discrete NVIDIA GPU: price the whole workstation

An RTX 5060 Ti has a 16GB option; the RTX 5070 Ti has 16GB and RTX 5090 has 32GB. They deserve evaluation when the application benefits from CUDA or dedicated GPU memory. VRAM remains a constraint, and ordinary system RAM does not become equally fast VRAM. No current complete-PC quote was verified here; advertised starting GPU prices should not be treated as the cost of a usable workstation. [27]

An AI badge is not a throughput test

Advertised TOPS or FLOPS depend on precision, sparsity and the component being measured. A runtime using the GPU does not automatically use the NPU. Confirm the actual backend and task speed instead of comparing an NPU rating with a GPU or whole-system marketing figure.

DGX Spark and larger shared-memory systems

The reviewed GB10 Spark configuration has 128GB shared memory, 273GB/s bandwidth and an Arm CPU. Check Arm dependencies, the desired serving stack and sustained performance. Its retrieved NVIDIA marketplace page provides partner links without a current US price, so it is **quote-only** in this comparison. Marketing FP4 compute does not establish response speed. [28] [29]

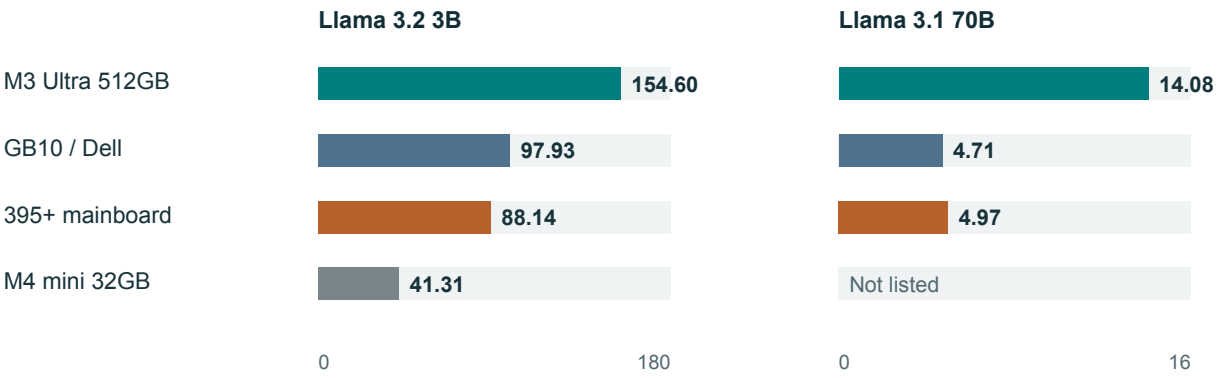
Recommendation: consider Spark for a validated NVIDIA development workflow, or a 192GB system for a demonstrated capacity need. Neither is the default budget choice without a complete quote and measured advantage.

09 / Published speed is useful context, not a buying rank

Historical measurements show why model size matters

The figures below are reported generation rates from Jeff Geerling's benchmark compilation, frozen at README revision `2f56f19`. They were not measured by ViWeb. The compilation spans different systems, software and run settings; it is not a controlled comparison of the current shopping list. [30]

Reported generation rates (tokens/s)



Different panel scales. Historical mixed-system reports; not a controlled procurement test.

Reported system	Llama 3.2 3B, tokens/s	Llama 3.1 70B, tokens/s
Mac Studio M3 Ultra, 512GB	154.60	14.08
Dell Pro Max with GB10	97.93	4.71
Framework mainboard, 128GB / 395+	88.14	4.97
Mac mini M4, 32GB	41.31	Not listed

The Framework 70B row is labeled GPU/CPU in the source, while its 3B row is GPU. The missing Mac mini 70B value is not zero. Model tags alone do not establish identical file hashes, cache settings or quantization. Follow the source's per-system links before making a closer comparison. [30]

What a purchase decision still needs

- **Prompt processing and time to first token:** long logs can be slow to ingest even when subsequent output looks fast.
- **End-to-end latency:** retrieval, queueing, tools and review contribute to the finished task.
- **Quality at the selected quantization:** faster output is of little use if it misses the important evidence.
- **Concurrent-user behavior:** one responsive session can coexist with poor latency for the rest of the team.
- **Steady-state energy and thermals:** adapter ratings, chip TDP and peak watts are not task energy.

The report intentionally does not divide these historical rates by today's prices to declare a winner. A defensible price-to-performance ratio needs the same task, model artifact, quality threshold and complete purchase cost. Page 12 gives a test that can produce that comparison.

10 / Value depends on accepted work and operating cost

Use two value measures, with a quality gate

Hardware value: accepted tasks per hour divided by the complete acquisition cost. Use this only among systems that pass the same task-quality and latency requirements. Report the actual task mix; a log summary and a code change are not interchangeable units.

Operating value: total monthly cost divided by accepted tasks completed that month. Include the review and correction time that makes an answer usable. Shared access, backup and support can create real costs even without per-token billing.

A basic cost model is:

```
Monthly cost = (purchase cost - residual value) / useful months
  + energy + administration + licenses + storage/backup
  + review/rework + other allocated costs
```

Illustrative scenario: utilization changes cost per task

This is transparent scenario arithmetic, not an ROI forecast, a current electricity tariff or a measurement of a listed device. Assume a $2,200 system, no residual value, 36 months of use, 100W average wall power while operating, 8 hours/day for 22 days/month and electricity at $0.15/kWh.

Monthly item	Calculation	USD
Hardware allocation	$2,200 / 36	$61.11
Energy	0.100kW x 176h x $0.15	$2.64
Administration	2 hours x $50/hour	$100.00
Subtotal for these three items	Unrounded calculation	$163.75
At 2,000 accepted tasks/month	Subtotal / 2,000	$0.082/task
At 200 accepted tasks/month	Subtotal / 200	$0.82/task

Licenses, backup, review, rework, taxes, financing and downtime are omitted from this subtotal, so it is **not full TCO**. At continuous operation for 30 days with the same assumed 100W draw, energy would be $10.80/month. Real idle and active power must be measured.

Compare local and hosted options on the same terms

Local is more attractive when utilization is sustained, the model meets the quality target and the organization values controlling sensitive processing. Hosted inference may be economically attractive for occasional or bursty work when data handling is acceptable. A hybrid design requires explicit routing so a failed local request cannot silently send confidential input elsewhere.

Test an existing device before replacing it. Power use alone cannot establish environmental benefit; hardware manufacture, utilization and the alternative workload also matter.

11 / A repeatable test before spending

Freeze the task and compare complete systems

Choose a small acceptance corpus that represents the intended work and has independently checked answers. Include benign cases, malicious cases and examples that should produce an uncertain answer. Use authorized or synthetic material; keep customer data within its agreed boundary.

Test layer	Record and compare
Identity	Exact device, purchase quote, memory, storage, OS, firmware, driver and power mode
Model	Publisher, immutable revision, file hash, tokenizer, quantization, context and cache settings
Inference	Runtime commit/version, backend, offload, prompt template, sampling and accelerator residency
Performance	Cold/warm startup, prompt rate, output rate, full-task p50/p95 latency, memory and wall energy
Service	One user and intended concurrency, queueing, timeouts, recovery after an error and sustained cooling
Quality	Factual support, missed threats, false alarms, valid queries, correctness and reviewer rework
Boundary	Cross-user isolation, hostile-document behavior, denied tool calls, egress and data deletion

Use a microbenchmark and an application replay

The upstream `llama-bench` tool supports prompt-processing and generation tests, repeated runs and machine-readable output. Start with a common artifact and configuration, then replay complete representative tasks. Native platform optimizations can be tested separately and labeled accordingly. [31]

Proposed pilot procedure: warm the system consistently, run at least five repetitions per selected test, and retain raw results. Exercise the expected context lengths and a sustained session long enough to reveal throttling. Report medians and spread; a five-run sample cannot support a dependable tail-latency claim. Use a longer application replay for p95 under realistic arrival patterns.

Set gates before viewing the results

A useful starting rule is: zero observed cross-customer disclosures or unauthorized actions in the defined adversarial set; no material loss in threat recall relative to the checked baseline; and a pre-agreed improvement in completed-task cost or latency. These are proposed acceptance conditions, not proof of zero real-world risk.

Document the sample size, counts and uncertainty. Review serious false negatives individually. A good score on a small corpus is a reason to expand a supervised pilot, not permission for unrestricted automation.

Buy the least expensive complete system that passes the quality, capacity, latency, security and support requirements. If none pass, change the workflow or model before increasing the hardware budget.

12 / Three practical cybersecurity workflows

A. Explain an unusual web-traffic pattern

A technician exports a bounded, authorized log sample and preserves the original. Deterministic queries establish counts, time ranges, response codes and request paths. The assistant groups patterns and proposes explanations with record references.

The technician checks those explanations against raw requests, server configuration and known maintenance events. A traffic spike becomes a finding only when the evidence supports it. Blocking a source or changing a rule goes through the normal operational process. The model's prose is never the sole reason to alter production traffic.

Success measure: time to a correct, supported explanation, including verification; track missed malicious cases and false alarms. Keep attacker-controlled strings in the sample as data and include prompt-injection cases during evaluation.

B. Review a suspicious PHP or JavaScript change

Work from an authorized copy in a test environment. Compare with a trusted upstream version where available; run conventional static checks and inspect behavior. Ask the assistant to explain unfamiliar control flow, identify candidate risks and propose a narrow patch.

Verify paths, dependencies and any cited vulnerability independently. Test the patch and check regressions before deployment. Treat generated package names and installation commands as untrusted until confirmed. Do not run an unknown sample on the analyst's normal machine. [9]

Success measure: independently confirmed findings and safe fixes per hour, with correction effort included. An assistant declaring a file clean is not evidence of complete malware removal.

C. Draft an incident or recovery report

Provide a verified event timeline, scoped observations and approved remediation records. Ask for a draft that separates observation, inference and unknowns. Require each material statement to point back to an evidence item.

A technician verifies the affected systems, dates and actual test results. If a patch was applied but no full forensic review or restore test occurred, the report must say exactly that. Customer reports receive the same confidentiality treatment as their underlying evidence.

Success measure: time to a publishable, accurate draft and the number of substantive corrections. Do not count fluent writing as investigation success.

Keep the operating team capable

Rotate who verifies model-assisted work and retain exercises that require reading raw evidence. A model outage should leave technicians able to investigate, communicate and recover using the established tools. Record where the assistant helps, where it distracts and where its errors are costly. These workflow examples are proposals, not descriptions of completed ViWeb customer work.

13 / A staged adoption decision

Stage 1: qualify one use case on existing hardware

Choose a read-only workflow with a clear answer key, such as a runbook question set or bounded log summaries. Name an operator and define the data, permissions, evidence requirements and stopping rules. Record an unassisted baseline and evaluate the proposed assistant against it.

Exit condition: the assistant produces useful, verifiable outputs on the defined sample, and its failures are understood. Stop if the required model cannot meet the quality bar; a hardware upgrade alone does not correct factual or authorization failures.

Stage 2: compare only the hardware that solves a measured limit

Observed constraint	Candidates worth testing	What would justify spending
Small model fits; one technician needs a compact host	Existing machine or Mac mini in the tested capacity tier	Lower completed-task cost or sufficient responsiveness with acceptable support
Working set exceeds a small shared-memory device	Max+ 395 with 64GB; larger-memory Mac mini quoted separately	Measured fit and latency with OS and cache headroom
Model plus context exceeds the 64GB tier	Max+ 395 with 128GB; quote a comparable larger Mac	Useful quality improvement at a supportable total cost
The selected stack requires CUDA	Quoted discrete-GPU workstation or GB10 system	Verified software compatibility, sufficient VRAM/shared memory and acceptable task throughput
Working set exceeds 128GB	Quoted larger-memory Mac or 192GB platform	A task benefit that outweighs the capacity premium and operating burden

These are conditional shortlists. No row asserts the fastest device or a guaranteed model size. Keep quotes for memory upgrades and comparable storage with the evaluation record.

Stage 3: move from a device to a managed service

Before sharing an assistant, add identity controls, customer separation, quotas, protected transport, monitoring and a recovery route. Re-run the boundary and load tests after material model, runtime or tool changes. Maintain a known working configuration and a documented way to disable tools or return to ordinary workflows.

Introduce automation gradually: draft only, then narrowly scoped read-only tools, then reviewed changes if the evidence supports them. High-impact actions still need enforceable authorization and a clear operational owner.

Recommended default for ViWeb readers: start with one small supervised use case and the lowest-cost suitable existing device. Obtain a complete quote and run the page 12 test before treating a new Mac, Ryzen AI Max or GPU workstation as a value purchase.

14 / Evidence, limits and terminology

What this edition establishes

This is a desk-researched technical guide with sources checked on **17 November 2026, Asia/Ho_Chi_Minh**. It combines primary documentation, vendor specifications and listings, original research, and an identified historical benchmark compilation. Recommendations, proposed workflows and numerical scenarios are labeled as such.

It contains no ViWeb device measurements, customer productivity data, measured breach reduction or controlled comparison of the current shortlist. This prevents a universal speed-per-dollar ranking. No orders or checkout stock checks were completed.

Prices are a US baseline in USD. Apple MSRP evidence is dated September 2026; a checked-as-of date does not turn an older price list into a live offer. Promotions, memory variants, taxes, regional warranty, import costs and availability can change the result. Obtain a current delivered quote before purchase.

How to read the evidence

Evidence class	Appropriate use	Limit
Official security guidance	Identify threats and design controls	Does not certify this design or prove residual risk is zero
Vendor specifications and prices	Establish a named configuration and displayed amount	Does not establish actual task speed, full TCO or delivery
Original experiments	Understand results for the studied people and workflow	Does not establish the same effect on a local model
Historical benchmark reports	Illustrate measured behavior in the reported systems	Different runs and software limit causal comparison
Arithmetic scenarios	Explore memory or cost under explicit assumptions	Are not observations, market forecasts or performance promises

Terms used in this report

Inference: using an existing model to generate an answer. **Open-weight:** model parameters are available under a stated license; this does not by itself mean unrestricted use or a fully open training process. **Quantization:** storing or computing values at lower precision. **Prefill:** processing the input prompt. **Decode:** generating subsequent tokens. **KV cache:** saved attention state used during generation. **TTFT:** time to first token. **TCO:** total cost of ownership. **RAG:** retrieval-augmented generation.

Editorial disclosure

Prepared for ViWeb Technology with AI-assisted research, drafting and layout. Sources and arithmetic were checked during preparation; there was no separate human technical review in this workflow and no ViWeb laboratory test. This research edition is intended for public use; its recommendations remain subject to the evidence limits above. It does not claim the site's separate automated briefing workflow was used.

Check model licenses and customer data-handling terms for the intended use. References follow in first-citation order. Send corrections with supporting evidence through ViWeb's contact page.

References

All sources checked 17 November 2026. Numbers follow first citation. Vendor pages and documentation can change; dates below preserve the evidence type.

[1] Generative Artificial Intelligence Profile (NIST AI 600-1)

NIST | 26 July 2024; publication page updated 8 April 2026

Voluntary risk framework; supports confabulation, privacy and oversight concerns. It does not certify a local deployment.

[2] LLM01:2025 Prompt Injection

OWASP Gen AI Security Project | 2025 edition; live page

Direct and indirect injection; RAG and fine-tuning do not eliminate the problem.

[3] LLM08:2025 Vector and Embedding Weaknesses

OWASP Gen AI Security Project | 2025 edition; live page

Permission-aware retrieval, source integrity and cross-context leakage.

[4] Randomized Controlled Trials for Phishing Triage Agent

James Bono, Microsoft; arXiv preprint v1 | 17 November 2025

167 recruited analysts; 13 excluded; curated email queues and simulated agent errors. Product-specific results are not evidence of local-model performance.

[5] Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity

METR | 10 July 2025

16 developers and 246 tasks; historical coding study. Use with its 2026 follow-up, not as a current universal estimate.

[6] We are Changing our Developer Productivity Experiment Design

METR | 24 February 2026

Later experiment faces selection effects. Does not establish a clean, current population-wide productivity estimate.

[7] Annual Review 2026: The impact of AI on cyber threat

UK National Cyber Security Centre | 2026; review covers 1 September 2025 to 31 August 2026

Intelligence assessment, not a measured global incident rate or local-AI attribution study.

[8] LLM06:2025 Excessive Agency

OWASP Gen AI Security Project | 2025 edition; live page

Limit tools, permissions and autonomy; enforce downstream authorization.

References / continued

[9] LLM05:2025 Improper Output Handling
OWASP Gen AI Security Project | 2025 edition; live page
Generated output needs ordinary validation, contextual encoding and safe execution boundaries.

[10] Transformers Security Policy
Hugging Face | Live repository documentation
Prefer safetensors, pin reviewed revisions and inspect any code enabled through trust_remote_code.

[11] LLM03:2025 Supply Chain
OWASP Gen AI Security Project | 2025 edition; live page
Models, adapters, data, runtimes and dependencies extend the supply chain.

[12] LLM10:2025 Unbounded Consumption
OWASP Gen AI Security Project | 2025 edition; live page
Bound requests, resource use, queue depth and tool loops.

[13] Deploying AI Systems Securely
NSA, CISA, FBI and international partners; FBI-hosted copy | April 2024, version 1.0
Deployment security guidance; supports hardening, isolation, monitoring and incident response. Not a compliance certificate.

[14] llama.cpp
ggml-org | Live project documentation
Local inference with multiple acceleration backends. Backend support does not establish identical speed or feature parity.

[15] MLX documentation
Apple ML research | Live documentation
Apple-silicon machine-learning framework, unified memory and LLM examples.

[16] Ollama Local FAQ
Ollama | Live documentation
Local-only settings, restart behavior, localhost binding and GPU residency checks. Settings must match the installed release.

References / continued

[17] AMD RDNA3.5 system optimization

AMD ROCm documentation | Live documentation

Ryzen APU memory configuration and Linux kernel requirements; inspect the matching compatibility matrix before procurement.

[18] NASPO PSS: Apple-branded price list

Apple US | September 2026; page 1

Use Consumer (MSRP) column, not discounted contract prices. Supports named stock configurations; not a checkout or stock guarantee.

[19] Framework Desktop DIY configuration: Ryzen AI Max 300

Framework US configurator | Live configuration page

395/64GB system $1,959; 395/128GB system $3,449. Storage and OS extra; selection availability not independently confirmed.

[20] EVO-X2 Ryzen AI Max+ 395 mini PC

GMKtec US store | Live listing

Selected 64GB/1TB variant $2,199.99. Do not attach this headline price to 128GB variants; page reports an unavailable variant.

[21] MINISFORUM MS-S1 MAX 128GB Max AI Compute Edition

MINISFORUM US store | Live listing

128GB/2TB displays $3,959 and $4,949 with a sale-ended message. Conflicting promotional signals: quote required before purchase.

[22] Beelink GTR9 Pro Ryzen AI Max+ 395

Beelink store, USD listing | Live listing

128GB soldered LPDDR5x plus 2TB SSD: displayed sale price $4,199, comparison price $4,699; final stock and total unverified.

[23] EVO-X5 Pro Ryzen AI Max+ PRO 495

GMKtec US store | Live listing

Selected 192GB/2TB variant displays $6,799. Vendor says in stock; fulfillment not verified.

[24] Mac mini technical specifications

Apple US | Live product page

Current page lists M6 and M5 Pro, memory options and bandwidth. Not M4 benchmark evidence.

References / continued

[25] AMD Ryzen AI Max+ 395 specifications

AMD | Live product page

128GB maximum system memory, LPDDR5x-8000, 256-bit interface and Radeon 8060S.

[26] Ryzen AI Max+ 395: Breakthrough AI Performance in Thin and Light

AMD vendor blog | 2025

Vendor describes up to 96GB graphics allocation in 128GB configurations. This is not guaranteed application-usable memory on every OEM or OS.

[27] GeForce RTX 50 Series specifications

NVIDIA US | Live product page

RTX 5060 Ti includes a 16GB option; RTX 5070 Ti has 16GB and RTX 5090 has 32GB. Advertised starting prices are not current complete-PC quotes.

[28] DGX Spark hardware overview

NVIDIA | Live user guide

GB10, Arm CPU, 128GB unified memory, 273GB/s bandwidth and power specifications. Marketing compute is not measured token throughput.

[29] DGX Spark US marketplace listing

NVIDIA | Live product page

128GB/4TB listing redirects to authorized partners and exposes no current US price in the retrieved page. Treat as quote-only.

[30] AI/LLM benchmarks: Findings

Jeff Geerling and repository contributors | README snapshot 22 September 2026; individual runs older

Historical mixed-system compilation, not a matched November 2026 procurement test. Follow per-system issue links for settings; do not compute a purchasing rank from it.

[31] llama-bench performance testing tool

ggml-org | Live project documentation

Prompt-processing and generation tests, repetitions and machine-readable output. An application replay remains necessary.