

Local AI on Apple Silicon: From Hardware Capability to Enterprise Decision

How unified memory and software tools change the options for enterprise inference

By Dr. Jeffrey Allan · Eldris Research
Enterprise AI Adoption Gap · Research brief 11
October 5, 2026

© 2026 Eldris Inc. All rights reserved.

Executive brief

Apple silicon makes the employee's computer a credible place to evaluate language-model inference. Its shared CPU/GPU memory architecture, substantial memory configurations and supporting software let developers build applications that perform suitable AI work locally. The enterprise implication is a choice about where work runs, what dependencies it creates and who operates it.

The evidence supports that deployment opportunity. It does not establish that a Mac replaces every hosted model, that local inference is automatically cheaper or that an application is private simply because its model runs on the endpoint.

The decision is whether a defined workflow delivers acceptable quality, responsiveness and operating cost on representative hardware within an approved data boundary. This report addresses U.S. enterprise technology and workflow leaders considering an employee-facing assistant that drafts summaries of internal documents, with a person reviewing each summary.

Three findings guide that decision:

- **Shared memory changes a resource constraint.** CPU and GPU access to one pool can simplify model execution compared with a conventional system that has physically separate system and graphics memory. It does not eliminate memory limits or performance constraints.
- **Software makes the capability usable.** MLX and model-serving tools provide a route to selected third-party models; Apple's Foundation Models framework provides another application route. Model choice, device support and execution location still require verification.
- **Value depends on the complete workflow.** Local execution can remove a remote inference dependency for eligible tasks. Quality failures, review effort, endpoint contention and support can offset that benefit.

The practical recommendation is to evaluate an existing capable endpoint before purchasing specialized hardware, where the workflow and policy permit. Compare it with a credible hosted alternative using accepted work, service time and complete operating costs.

What changed in the deployment architecture

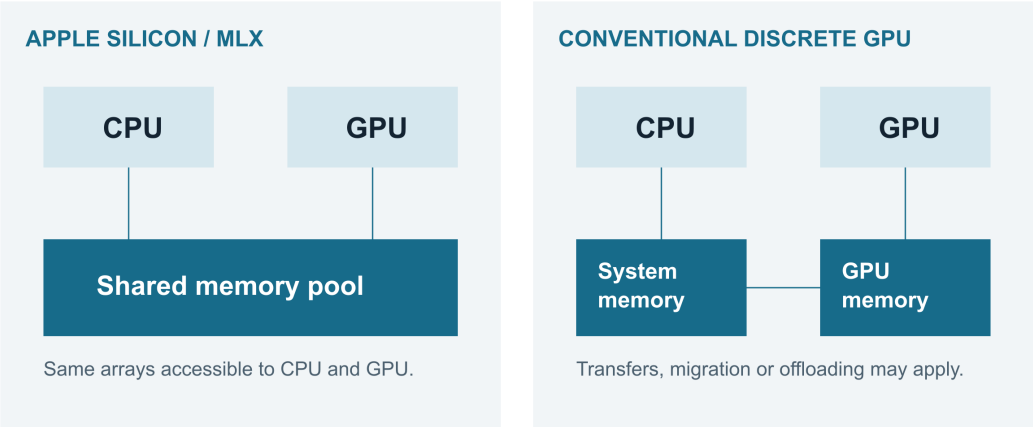
A model needs space for its stored parameters, execution buffers and conversation state. It also needs an efficient path between that memory and the processors doing the work. Having enough memory somewhere in a computer is not sufficient to ensure usable inference.

MLX's unified-memory documentation states that Apple silicon's CPU and GPU have direct access to the same memory pool. MLX operations can use the same arrays on either processor without relocating those arrays between separate CPU and GPU memory locations. [MLX documentation](#).

In a conventional discrete-GPU computer, system RAM and GPU memory are distinct resources. Transfers, managed migration or offloading can make a larger workload possible, but the arrangement must account for those resources and their connecting paths. NVIDIA's CUDA guide describes managed-memory migration, oversubscription and hardware-dependent behavior. A common address space should not be mistaken for an identical physical memory architecture. [CUDA guide](#).

Shared memory changes a deployment constraint

Architecture schematic · Not a performance comparison



A shared pool is still finite: operating system, applications and inference compete for it. Other platforms can also use coherent unified memory; this is not an Apple-only concept.

Original Eldris schematic. Sources: MLX Unified Memory; NVIDIA CUDA memory guide.

© Eldris Inc

Figure 1. Original Eldris schematic based on MLX and CUDA documentation. The discrete-GPU arrangement is a conventional example, not every non-Apple platform. The diagram establishes no performance ranking.

The strategic benefit is that a local application can use a substantial shared memory pool without requiring a separate graphics card with its own capacity limit. This can simplify one part of deployment. The operating system, display and ordinary applications still consume resources, and the inference software still determines how efficiently the machine uses them.

Apple is not the only supplier of a coherent shared-memory system. NVIDIA currently documents DGX Spark with coherent unified memory configurations of 64 GB or 128 GB. That is a purpose-built alternative to evaluate, not evidence that Apple invented local AI or that one architecture wins every workload. [NVIDIA specifications](#).

Decision implication: Treat Apple silicon as an additional deployment option. Compare the actual machine, runtime and workload rather than turning the memory architecture into a universal platform recommendation.

Capacity and bandwidth answer different questions

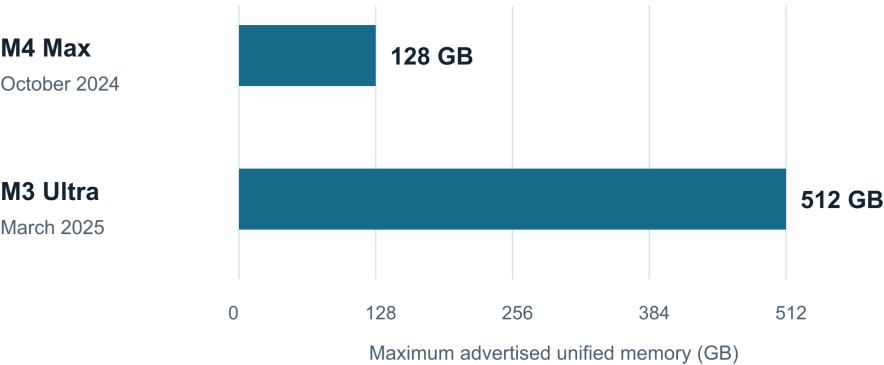
Memory capacity helps determine what can reside in memory. Bandwidth helps determine how quickly data can move through the memory system. Neither figure independently establishes the time an employee waits for a usable answer.

Apple's October 30, 2024, M4 Max announcement specified support for up to 128 GB of unified memory and up to 546 GB/s of memory bandwidth. The corresponding 2024 16-inch MacBook Pro specifications identify configuration differences: the 14-core CPU/32-core GPU M4 Max has 410 GB/s, while the 16-core CPU/40-core GPU variant has 546 GB/s. The 128 GB option belongs to the latter configuration. [Announcement](#), [configuration specifications](#).

Apple's March 5, 2025, M3 Ultra announcement specified configurations up to 512 GB of unified memory and over 800 GB/s of memory bandwidth. These are historical manufacturer specifications, not a statement about current inventory, procurement price or the configuration of an ordinary employee's Mac. [M3 Ultra announcement](#).

Published capacity is not measured performance

Selected historical Apple specifications · Configuration maxima



BANDWIDTH, A DIFFERENT RESOURCE

M4 Max: up to 546 GB/s

M3 Ultra: over 800 GB/s

M4 Max values refer to the highest specified variant, not every M4 Max machine.
These are not model memory allowances, current inventory or inference benchmarks.

Sources: Apple M4 Max announcement and 2024 MacBook Pro specs; M3 Ultra announcement.

© Eldris Inc

Figure 2. Selected historical manufacturer specifications. Bars show maximum advertised memory capacity, not measured inference performance or memory available to a model. Bandwidth values are listed separately because they describe a different resource.

These configurations demonstrate a range of local resource options, including a laptop and a high-memory desktop. They do not demonstrate the same user experience. A machine that holds a model may still process long documents too slowly, contend with other applications or fail the task’s quality requirement.

Measure document processing, time to first response, complete response time and review effort separately. A high token-generation rate after processing a short prompt can conceal a long wait to read the actual document. A faster answer that requires more correction can reduce business value.

Decision implication: Inventory exact fleet configurations. Test an employee's ordinary workload alongside the assistant; do not use the highest advertised chip specification as the baseline for the fleet.

Model size is only the start of memory planning

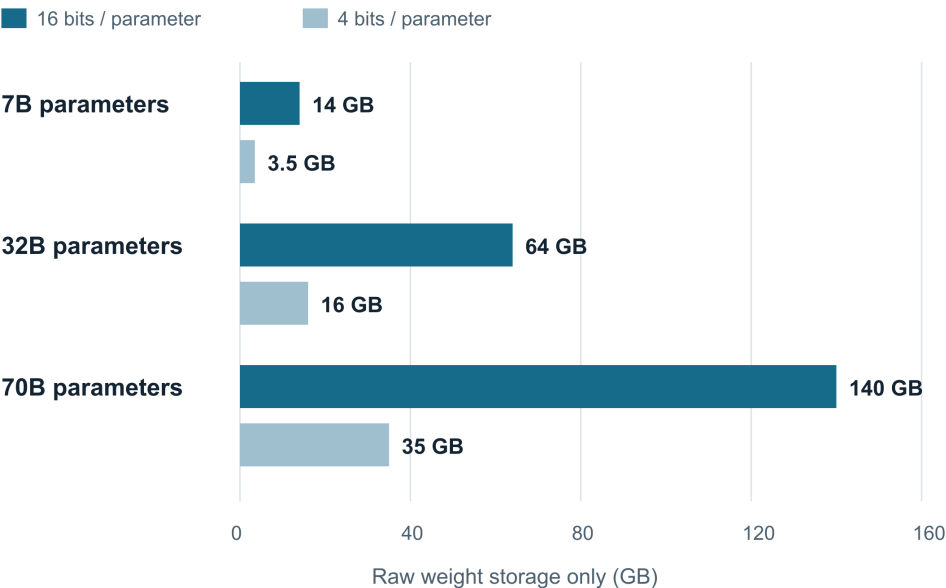
A useful first approximation for weight storage is:

Parameter count × bits per parameter / 8 = raw weight bytes.

For a hypothetical dense 32-billion-parameter model, storing every parameter at 16 bits requires 64 GB of raw weight data. At 4 bits, the arithmetic gives 16 GB. These use decimal GB, where 1 GB equals one billion bytes. They are calculations, not measurements of a model file or a deployed application's memory.

Weight storage shrinks with lower precision

Hypothetical dense models · Raw weight arithmetic · Decimal GB



Formula: parameter count × bits per parameter / 8. Assumes uniform storage.
Excludes quantization overhead, cache, buffers, applications and operating system.
Actual memory use and quality must be measured for the selected model and runtime.

Original Eldris arithmetic. A smaller raw weight total does not establish model fit.

© Eldris Inc

Figure 3. Original Eldris arithmetic for hypothetical dense models. It assumes uniform 16-bit or 4-bit storage for all parameters. Actual formats can include mixed precision, scales, metadata and other overhead. These bars exclude runtime memory and do not establish model fit or quality.

Quantization can reduce storage requirements. It can also change quality and runtime behavior, so the compressed model needs its own task evaluation. A smaller file is not proof that the assistant preserves important facts in a document summary.

The full memory plan includes working buffers, model state associated with the prompt and generation, the retrieval index if used, the application and other workloads. MLX LM's documentation describes configurable key-value caching and prompt-processing steps, including tradeoffs among memory, processing speed and quality. It also warns that models large relative to system RAM can be slow. [MLX LM documentation](#).

Sparse architectures add another distinction. Total stored parameters and parameters active for a request need not be the same. Apple's June 8, 2026, account describes AFM 3 Core Advanced as a 20-billion-parameter model with 1 to 4 billion parameters active, using a design that keeps the full model in flash and selectively loads experts into memory. This is Apple's description of a particular architecture, not a rule for arbitrary open-weight models. [Apple research](#).

Decision implication: Select the model, numerical format, context requirement and runtime together. Validate peak memory and task quality instead of estimating deployability from parameter count alone.

The application layer is part of the evidence

Hardware becomes an enterprise option when software can turn it into reliable work. Two distinct routes illustrate that development opportunity.

MLX LM documents text generation, quantization and fine-tuning for language models on Apple silicon. Its supported-model and memory guidance provides an implementation route for selected third-party weights. It does not guarantee the license, suitability or behavior of every model someone can download. [MLX LM](#).

Apple's developer page describes Foundation Models as a native Swift API with access to Apple models on-device and in Private Cloud Compute, as well as conforming third-party providers. The important enterprise question is which model and execution route the application actually uses. Calling a framework associated with on-device AI does not establish that every request remains local. [Apple developer resources](#).

Apple's 2026 research account also describes separate on-device and server models. That is evidence of a differentiated deployment strategy, not a claim that all work can move to the endpoint. The report does not infer general third-party access to every model in that family from the research announcement.

An application must handle device and model availability, document size, unsupported requests, cancellation and resource contention. It should explain where work runs and obtain the appropriate authorization before sending local documents to an optional hosted route. If local processing cannot complete the task, silently changing the data boundary is not an acceptable recovery design.

For the summary assistant, the interface should make source documents and review obligations clear. Administrators need to know which model version is deployed, how updates are controlled and what information the application stores. Employees should not need to manage GPU memory to complete a routine task.

Decision implication: Evaluate the complete application. A working model demonstration establishes a technical possibility; a supported, understandable workflow establishes a deployment candidate.

Local execution changes the data path, not every privacy obligation

When inference and its necessary inputs remain on the endpoint, those inputs need not be sent to an external inference service for that operation. This is an architectural opportunity to reduce a data transfer. It must be verified in the actual product.

Trace the complete summary workflow: document retrieval, text extraction, model execution, saved output, diagnostics, backups, search and optional actions. A locally executed model can sit inside an application that sends documents, prompts or diagnostic content elsewhere. A cloud-hosted document repository is also a distinct data path even if inference happens locally.

Document what remains on the endpoint, what leaves it and why. Test the configured workflow rather than relying on an application label. Check retained local records, access by other users and administrators, and how sensitive work is deleted or recovered. A boundary is only useful if the organization can operate it.

Local execution does not establish compliance, neutralize malicious inputs or make a generated summary accurate. NIST's Generative AI Profile calls for deployment-relevant evaluation and examination of security and privacy risks; it does not exempt systems that run on endpoints. [NIST profile](#).

Decision implication: Approve a verified data flow and operational arrangement. Specify any permitted hosted fallback, its data categories and who can authorize it.

Compare cost per accepted outcome

Local inference can avoid a hosted inference charge for work that it successfully completes without calling that service. It still consumes hardware capacity, electricity, software and support. The comparison should include correction and employee time, not just a provider's token price.

Separate two decisions. Using an existing capable Mac creates an incremental deployment case: what additional software, support, energy and opportunity costs does the workflow introduce? Purchasing a new high-memory Mac creates an acquisition case that also includes capital cost, utilization, useful life and any accelerated replacement of existing equipment.

A previously purchased computer should not be charged its full purchase price again as a new cash outlay. It should not be treated as having unlimited free capacity either. Report the incremental cash comparison and any allocated economic cost clearly; avoid counting the same hardware burden twice.

For each option, calculate **complete workflow cost divided by accepted outcomes** over the same period and at the same quality requirement. Include software, integration, administration, energy, review, rework and failures. Include the relevant hardware burden for local options and the actual service billing arrangement for hosted options.

A hosted subscription may not fall when some requests move locally. A committed contract may leave the provider bill unchanged. Measure avoidable spending instead of multiplying diverted tokens by a public list price. A smaller hosted model or improved workflow may be a better alternative than the most expensive API.

Local capacity can be stranded across many endpoints, while a shared service can pool demand. Conversely, recurring work on already available endpoints may justify keeping suitable inference there. Document demand, simultaneous use and the cost of delays. No universal break-even volume follows from the public sources in this report.

For the summary assistant, more accepted summaries at the required quality can be valuable; a larger number of generated summaries is not sufficient. If users spend the saved production time correcting omissions, the apparent inference saving can disappear.

Decision implication: Build the economic case with observed accepted work and actual avoidable costs. Quantify quality and review differences before claiming savings from local execution.

Choose the deployment that passes the workflow requirements

Choose where the workflow should run

Apply the same quality requirement to every deployment option

OPTION TO EVALUATE	CONDITIONS TO TEST	BUSINESS EVIDENCE
EXISTING ENDPOINT	Task quality, resource headroom and support requirements pass.	Evaluate incremental cost and approved data paths.
DEDICATED LOCAL	Local task is suitable; endpoint capacity or contention fails.	Test added capacity, utilization and acquisition economics.
HOSTED SERVICE	Capabilities or service capacity better meet the requirement.	Verify permitted data transfer and complete workflow costs.
HYBRID ARRANGEMENT	Different tasks need different approved execution routes.	Make routing, authorization and fallback explicit.

If no option passes, narrow the task or retain the current process.
Original Eldris guidance; not measured adoption rates or a platform ranking.

© Eldris Inc

Figure 4. Original Eldris decision guide. The branches are evaluation paths, not measured adoption rates or a prediction of which platform wins. Quality and policy requirements apply to every option.

An existing endpoint is a candidate when the selected model meets the task requirement, ordinary work remains responsive and the application can be supported within an approved boundary. This is a useful first test where it can answer the decision without a hardware purchase.

A dedicated local machine is a candidate when endpoint contention or capacity prevents suitable local work and the business value justifies additional equipment and operation. It introduces a shared resource to manage. A desktop serving a team is a different capacity proposition from an employee running an assistant alone.

A hosted service is a candidate when its capabilities, elastic capacity or operating arrangement better meet the requirement and the data flow is permitted. Remote infrastructure remains real even when a service makes it easy to consume.

A hybrid arrangement is a candidate when routing can preserve quality and enforce the permitted data boundary. Document which work is local, which work may be remote and what happens when the route is unavailable. Do not assume every local request has an approved cloud fallback.

If none passes, narrow the task or retain the current process. The purpose is a better business outcome, not maximizing the percentage of requests labeled local.

Decision implication: Make the choice at the workflow level. Local and hosted deployment can coexist under explicit rules rather than a single company-wide assumption about where AI belongs.

Evidence to collect before expansion

Begin with one document-summary workflow and representative hardware groups. Record chip variant, installed memory, operating-system version, runtime, model artifact, quantization, configuration and application version. Use approved test documents that reflect the length, structure and ambiguity of the intended workload.

Establish the current process and a credible hosted comparison. Use a consistent evaluation rubric: preservation of material facts, consequential omissions, unsupported statements, source traceability and the effort needed to reach an acceptable summary. Include difficult cases and documents that should be rejected or escalated.

Measure cold and warm starts, prompt processing, complete response time, peak memory and impact on ordinary applications. Test normal power conditions and sustained use where relevant. Report completion and failure rates as well as medians and slow-case service times. A few short prompts on an idle machine do not qualify a fleet.

Use blinded quality review where feasible. Keep reviewers unaware of which deployment produced the summary and record their correction time. When a model or quantization changes, reassess the output rather than transferring a result from another artifact.

Verify the configured data paths, update controls and recovery behavior. Gather support effort during real use. Predefine acceptable quality and service requirements, along with what would stop or narrow the deployment. Report uncertainty and hardware-group differences.

NIST recommends evaluation in conditions similar to deployment and cautions against extrapolating capabilities from narrow anecdotal assessments. This report's test plan applies that principle; it does not claim an enterprise validation study has already been conducted. [NIST guidance](#).

The resulting approval record should identify the permitted workflow, tested configurations, accountable owner, observed costs and conditions requiring reassessment. That evidence can support using existing endpoints, buying additional local capacity or selecting a hosted option.

The enterprise implication

Apple's contribution is concrete: a documented shared-memory architecture, high-capacity configurations and software routes for building AI applications. These support treating the Mac as a location for suitable inference, not merely an interface to a remote service.

The proposition that Apple made local AI a software decision is therefore best understood as an enterprise opportunity, not a claim of historical exclusivity or universal readiness. Developers can package capable local workflows as applications. They still have to earn adoption through task quality, resource management, clarity and support.

For enterprise leaders, the actionable conclusion is to include local Apple silicon execution in a bounded deployment comparison when fleet conditions and workload requirements make it plausible. Require the same evidence of accepted outcomes from local and hosted options. Buy additional hardware only when the measured workload justifies it.

Apple expanded the deployment choice. Software and operating evidence determine whether the enterprise should use it.

Method, limitations and disclosure

This report reviews primary sources checked on October 5, 2026: architecture and runtime documentation, selected manufacturer specifications, Apple research and NIST guidance. It adds original Eldris analysis of memory arithmetic, deployment choices, economics and evaluation design.

Manufacturer specifications establish published configuration capabilities. They are not independent latency, throughput, power-consumption or cost measurements. The selected 2024 and 2025 hardware examples illustrate resource options; they are not a current purchasing guide or a comprehensive comparison of Apple chips. The 2026 model discussion establishes Apple's published architecture description, not third-party availability of every model or enterprise task performance.

No hardware benchmark, enterprise trial or survey was conducted for this report. It provides no quantified savings estimate, fleet qualification, regulatory clearance or platform-wide performance ranking. Arithmetic is labeled and excludes runtime overhead. Actual deployment depends on the model, software, documents, hardware configuration, concurrent work and operating conditions.

Eldris develops OOMU, an AI software product discussed in the originating opinion piece. The author has a commercial interest in local AI software. OOMU's development account is identified as such; this report contains no independent assessment of that product and implies no Apple endorsement or partnership.

This report builds on the author's October 5, 2026, opinion, Apple Made Local AI a Software Decision. Its public evidence supports a deployment opportunity and a decision process. Claims of enterprise value should follow measurements of the actual workflow.

Sources

1. **MLX Contributors. Unified Memory.** Documentation checked October 5, 2026. Shared CPU/GPU memory and array access on Apple silicon. [Documentation](#).
2. **NVIDIA. CUDA Programming Guide, Unified and System Memory.** Documentation checked October 5, 2026. Managed-memory migration, oversubscription and hardware-dependent behavior. [Guide](#).
3. **Apple. Apple introduces M4 Pro and M4 Max.** October 30, 2024. Published memory-capacity and bandwidth limits. [Announcement](#).
4. **Apple Support. MacBook Pro (16-inch, 2024) - Tech Specs.** Checked October 5, 2026. M4 Max variant-specific bandwidth and configurable memory. [Specifications](#).
5. **Apple. Apple reveals M3 Ultra, taking Apple silicon to a new extreme.** March 5, 2025. Published memory capacity up to 512 GB and bandwidth over 800 GB/s. [Announcement](#).
6. **MLX Contributors. MLX LM.** Repository documentation checked October 5, 2026. Model generation, quantization, caching and memory guidance. [Repository](#).
7. **Apple Developer. Apple Intelligence.** Checked October 5, 2026. Foundation Models application interfaces and on-device, Private Cloud Compute and provider routes. [Developer resources](#).
8. **Apple Machine Learning Research. Introducing the Third Generation of Apple's Foundation Models.** June 8, 2026; page updated September 9, 2026. On-device and server model distinctions and sparse on-device architecture. [Research overview](#).
9. **NVIDIA. DGX Spark.** Product specifications checked October 5, 2026. Coherent unified memory options; a non-Apple example. [Specifications](#).
10. **NIST. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile, NIST AI 600-1.** July 2024. MEASURE 2.3, 2.5, 2.7 and 2.10 address evaluation, limits of generalization, security and privacy. [Profile](#).