

Intelligence at the Edge

The technical and economic case for
on-premise, on-device AI.

Cloud AI works, and the cost of it working is that your material has to travel to reach it. This paper sets out what the published research now says about that trade — what employees actually send to external models, what a breach of it costs, and how dependable the infrastructure underneath really is — and then specifies a complete architecture for running the same class of assistant entirely inside your own perimeter: model runtime, retrieval, workflow, guardrails, memory, integrations, interface and deployment, assembled almost entirely from MIT and Apache-2.0 licensed components.

What this paper covers

01 Executive summary	10 Memory — the scholar blueprint
02 The exposure problem, measured	11 The open-source foundation
03 The dependability problem	12 Hardware sizing and deployment tiers
04 The deployment spectrum	13 Economics and break-even
05 Reference architecture	14 Sector blueprints
06 The data pipeline	15 Honest limitations
07 Retrieval — RAG on private material	16 A 90-day implementation path
08 Reasoning models (RLM) at the edge	17 References
09 Guardrails and audit	18 Legal notice

WHO THIS IS FOR

CTOs and CISOs evaluating whether AI can be adopted without widening the attack surface; clinical and research IT leads under HIPAA, GDPR or equivalent data-residency obligations; heads of engineering costing per-seat AI against owned infrastructure; and technically-minded individuals who want a capable assistant that does not require them to upload their life.

HOW TO READ IT

Sections 2 and 3 are the evidence base and cite published third-party research. Sections 4 to 12 are architecture and can be read as an implementation specification. Section 13 is a cost model with its assumptions stated. Section 15 is the part most vendor papers omit — where this approach is genuinely worse than the cloud, and how to tell in advance whether that applies to you.

No IOT is pre-release. Every figure in this paper describes the published state of the industry, not the measured performance of a No IOT product.

Executive summary

The argument of this paper is narrow and structural: an assistant that answers where the question was asked has no transmission to intercept, no third-party retention policy to trust, and no shared failure domain to inherit. Everything else follows from that.

WHAT THE EVIDENCE SHOWS

- **Exposure is routine, not exceptional.** Research published in 2025 found that roughly three in four employees paste corporate data into generative AI prompts, and that a large majority of those pastes are made from personal accounts outside any company control.
- **The sensitive share is rising fast.** The proportion of corporate data entering AI tools that is classed as sensitive has more than tripled in two years.
- **It is already showing up in breach statistics.** IBM's 2025 study found unsanctioned "shadow AI" involved in one in five breaches, and that almost every organisation breached through an AI system lacked AI access controls.
- **Concentration is a reliability risk.** A single configuration fault in one cloud region in October 2025 degraded services worldwide for over fifteen hours. No attacker was involved.
- **Distance costs time.** Cloud inference carries a network round trip that on-device inference does not — a structural gap, independent of model quality.

\$4.44M Global average cost of one data breach. IBM · 2025 [1]	\$7.42M Healthcare average — costliest sector. IBM · 2025 [1]	77% Employees pasting data into AI prompts. LayerX · 2025 [3]	0 Packets leaving a device that runs its own model. By architecture
---	--	--	--

WHAT WE PROPOSE INSTEAD

A complete assistant stack that runs inside the customer's perimeter: a quantised open-weight model, a local retrieval index over the customer's own documents, a workflow engine for multi-step tasks, guardrails and an audit log that execute locally, a persistent memory the user owns, integrations into existing systems, a first-class interface, and a deployment path that survives having no internet connection at all. Section 5 specifies the layers; Section 11 shows that nearly all of them can be assembled from MIT and Apache-2.0 licensed components today.

Bring the intelligence to the data, instead of sending the data to the intelligence.

WHAT IT COSTS

On the illustrative model in Section 13 — fifty knowledge workers, one private inference node — owned infrastructure overtakes per-seat subscription cost at around month thirty-two. That is a slow payback, and we would rather state it plainly than tune the assumptions until it flatters us: the crossover arrives much earlier with headcount and heavy usage, and later with hardware ambition. Cost is not the main argument. The main argument is that some categories of data should not leave the building at any price.

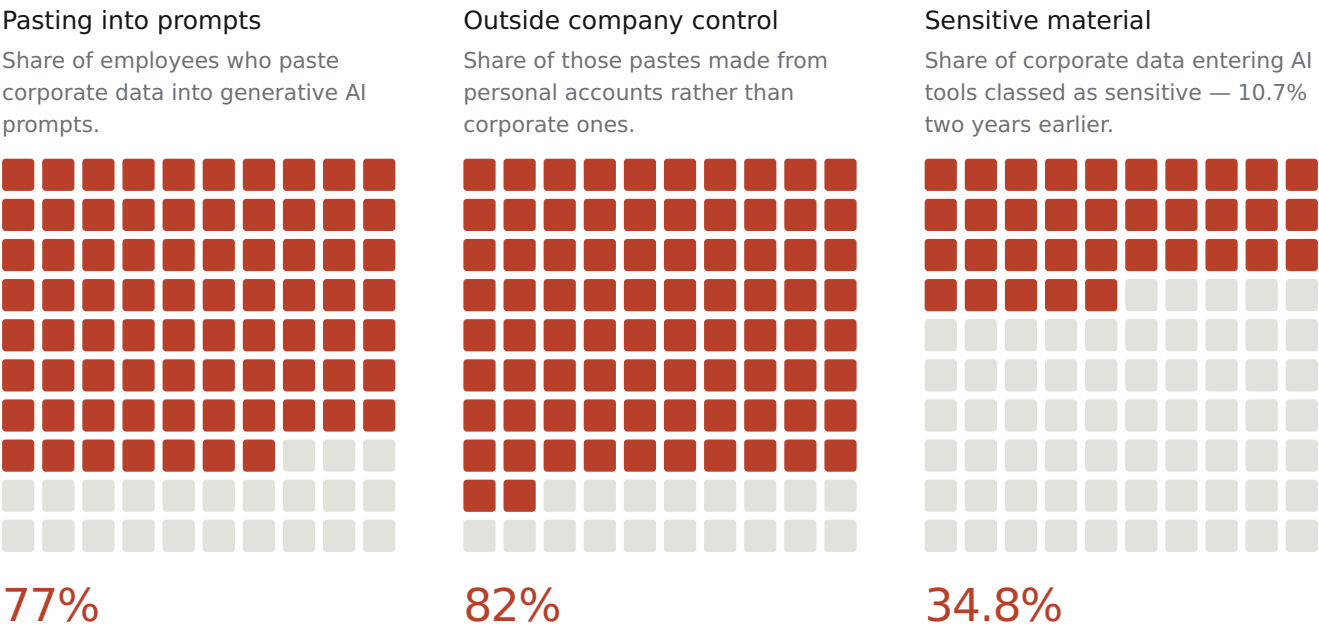
The exposure problem, measured

The risk of cloud AI is not that providers are careless. It is that using them requires a transfer of custody, and transfers of custody are where organisations lose control of material they are legally responsible for.

2.1 What people actually send

The gap between AI policy and AI behaviour is now well documented. Employees are not exfiltrating data maliciously; they are pasting a contract, a patient summary or a codebase into whichever assistant is open because it makes the afternoon shorter. The instrumentation research published through 2025 puts a number on how often that happens.

FIG. 1 · Employee data flowing into generative AI tools · each square = 1 in 100



Sources: LayerX, Enterprise AI and SaaS Data Security Report 2025 [3]; Cyberhaven, 2025 [4]. Reported averages run to roughly fourteen pastes per employee per day into non-corporate accounts, of which around three carry sensitive content.

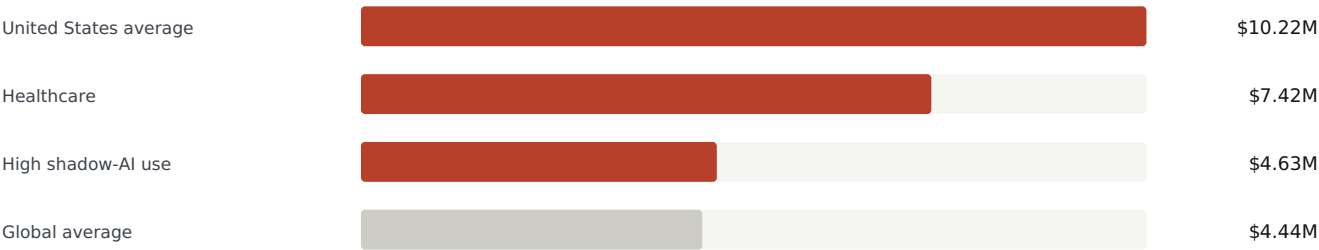
Two of those numbers matter more than the headline. The **82% from personal accounts** means the organisation has no log, no retention setting and no deletion route — the data is governed by a consumer

terms-of-service the employee accepted privately. And the **rise from 10.7% to 34.8%** is the trend line: as assistants get more capable, people entrust them with more consequential material.

2.2 What it costs when it goes wrong

IBM's annual study, conducted with the Ponemon Institute across 600 breached organisations in 17 industries and 16 countries, is the standard reference. The 2025 edition was the first in five years to show a global decline — driven by faster detection — while US costs reached a record.

FIG. 2 · Average cost per breach, 2025 (USD millions)



Source: IBM Security & Ponemon Institute, Cost of a Data Breach Report 2025 [1]. Breaches took an average of 241 days to identify and contain globally; healthcare averaged 279 days [1][5].

THE AI-SPECIFIC FINDINGS

- **20%** of breaches involved shadow AI — unsanctioned tools running without oversight.
- **97%** of organisations breached through an AI system had no AI access controls in place.
- **63%** had no governance policy covering AI use or shadow AI risk.
- **+\$670,000** average additional breach cost where shadow AI use was high.
- **13%** reported breaches of AI models or applications; a further **8%** did not know whether they had been.

IBM Cost of a Data Breach Report 2025 [1][2].

2.3 The custody chain

Each of those findings describes the same underlying shape. A cloud request crosses several administrative boundaries, and at every one of them the questions that matter — who may read this, how long is it kept, which jurisdiction governs it, who is notified if it leaks — are answered by somebody else's policy.

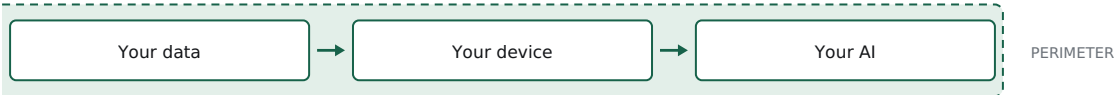
FIG. 3 · Custody points per request

CLOUD PATH — 6 CUSTODY POINTS



Each boundary is a place where retention, jurisdiction and access control stop being yours.

LOCAL PATH — 0 CUSTODY POINTS



Illustrative architecture comparison. The count is of administrative boundaries, not network hops; a single provider may itself involve several sub-processors.

The local path is not more secure because local software is better written. It is more secure because the set of parties who could possibly be compelled, breached or misconfigured has one member.

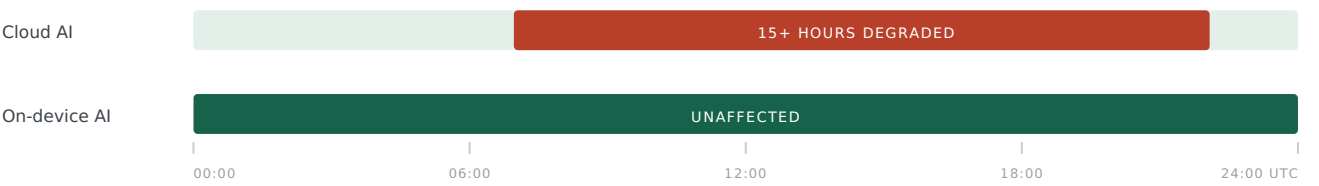
The dependability problem

Privacy is the argument people expect. Dependability is the one that tends to decide deployments, because it is the one that shows up on a Monday morning.

3.1 Shared failure domains

On 20 October 2025, a race condition in the DNS management system behind DynamoDB in a single AWS region cascaded into the services that a large share of the internet is assembled from. Downstream platforms — messaging, banking, airlines, retail, smart-home devices — failed together, because they had independently chosen the same dependency. There was no attacker; a routine technical update was sufficient.

FIG. 4 · 20 October 2025, us-east-1 — availability across a 24-hour window



Sources: Cisco ThousandEyes outage analysis, 20 October 2025 [6], which measured the disruption at over fifteen hours; AWS post-event summary; aggregate outage reports [7]. Reported impact spans 113 AWS services and 1,000+ dependent platforms, with more than 11 million outage reports logged globally.

The relevant property is not the outage length. It is that thousands of organisations discovered simultaneously that they had one failure domain rather than thousands. An assistant whose weights sit on the machine in front of the user does not participate in that correlation. It fails when that machine fails, which is a risk the owner can see, insure and mitigate.

3.2 Latency is structural

A cloud request pays for distance twice — once out, once back — before the model has produced a token. For chat that is an irritation. For a clinician dictating mid-procedure, a factory-floor vision check or a voice assistant expected to feel immediate, it is the difference between usable and not.

FIG. 5 · Time to first response, same task class



Sources: comparative edge-versus-cloud inference literature [8]; measured network latency to centralised model services ranges from roughly 48 ms to several hundred ms depending on provider and user location [9]. Figures vary widely with model size, quantisation and hardware — the gap is structural rather than incidental.

Offline capability is not a degraded mode. For an on-device system it is the normal mode, and the network is the optional extra.

3.3 What "dependable" means concretely

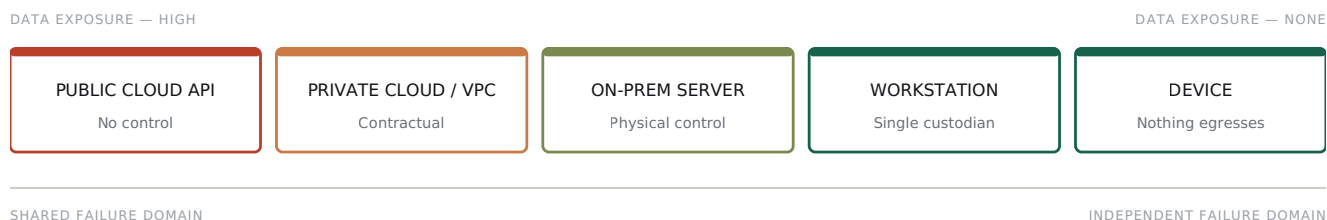
PROPERTY	CLOUD AI	ON-PREMISE / ON-DEVICE
Availability during ISP outage	None	Full
Availability during provider incident	None	Unaffected
Model behaviour stability	Changes when the provider updates	Frozen until you choose to update
Deprecation risk	Endpoint can be retired	Weights remain on disk
Rate limiting	Provider-imposed	Bounded by your own hardware
Cost predictability	Per token, variable	Capital cost, then power
Failure blast radius	Shared with every other tenant	One machine, one owner

Model-behaviour stability is frequently the deciding property in regulated settings: a validated clinical or legal workflow cannot silently change underneath its validation.

The deployment spectrum

"On-premise" and "on-device" are not one choice. They are a spectrum, and most organisations will sit at more than one point on it simultaneously.

FIG. 6 · Where inference can run, and what changes at each step



Moving right reduces both the number of parties holding your data and the size of the failure domain you share. It also reduces the maximum model size you can run — Section 12 sets out the trade.

READING THE SPECTRUM

- **Public cloud API.** Best raw capability, zero infrastructure work, and complete dependence on a third party for privacy, availability and continuity.
- **Private cloud / VPC.** Improves the contract, not the physics. The data still leaves your premises and still lives on hardware you do not control.
- **On-prem server.** The practical target for most organisations: a GPU node in your own rack serving the whole team, under your existing physical and network controls.
- **Workstation.** A single powerful machine for a department, a clinic or a research group. No new networking, no data-centre requirement.
- **Device.** Laptop, tablet or embedded unit. Smallest models, absolute privacy, works on a train and in a basement.

THE HYBRID PATTERN MOST TEAMS END UP WITH

Sensitive work — patient notes, contracts, unpublished research, source code — is answered locally and never transmitted. Non-sensitive work with no privacy dimension may still be routed to a frontier cloud model when raw capability matters more than custody. The important design property is that this routing is **explicit, logged and default-closed**: nothing leaves unless a policy says it may, and the audit log records that it did.

Reference architecture

A local model is not a product. The distance between "a model runs on this laptop" and "a clinician uses this daily" is eight layers, and seven of them have nothing to do with the model.

FIG. 7 · The on-premise assistant stack

08	DEPLOYMENT Installer, updates, device attestation, offline licence, rollback
07	USER INTERFACE Desktop, web on LAN, voice, mobile on private network
06	INTEGRATIONS Files, mail, EHR/LIS, ERP, calendars, cameras, sensors, MCP tools
05	MEMORY Working, episodic and semantic memory — local, exportable, erasable
04	GUARDRAILS Input policy, PII handling, tool allow-list, output checks, audit log
03	WORKFLOW ENGINE Planning, tool calls, multi-step orchestration, retries, human approval
02	RETRIEVAL (RAG) Chunking, embeddings, vector + keyword index, reranking, citation
01	MODEL RUNTIME Quantised open-weight LLM / RLM on CPU, GPU or NPU

Layers 01-05 are the engine; 06-08 decide whether anyone actually uses it. Most failed on-prem AI projects have a working layer 01 and nothing above it.

5.1 What each layer is responsible for

01 · MODEL RUNTIME

Loads quantised open-weight models and executes inference against whatever silicon is present — CPU, discrete GPU, integrated GPU or NPU. Responsible for memory mapping, context management, batching and concurrency. This layer determines what hardware the whole system needs.

02 · RETRIEVAL

Turns the customer's own documents into something the model can consult at answer time: chunking, embedding, a vector and keyword index, reranking, and citation of the passage a claim came from. Section 7.

03 · WORKFLOW ENGINE

Chat is one interaction pattern; most real work is a sequence. This layer plans multi-step tasks, calls tools, handles retries and failures, checkpoints long jobs, and pauses for human approval where policy requires it. It is what turns "summarise this" into "read the intake form, extract the medication list, cross-check it against the allergy record, and flag conflicts for the nurse to confirm."

04 · GUARDRAILS

Policy applied on the way in and on the way out, plus the audit trail. Section 9.

05 · MEMORY

Working memory within a session; episodic memory of what has been done; semantic memory of durable facts and preferences. Owned by the user, inspectable, exportable and erasable. Section 10.

06 · INTEGRATIONS

Adapters into the systems where work already lives: file shares, mail, calendars, EHR or LIS, ERP, ticket systems, cameras and sensors, and standard tool protocols. Every adapter is an explicit grant, not an ambient permission.

07 · USER INTERFACE

Desktop application, a web interface served on the local network, voice, and mobile clients that reach the on-prem node over the private network only. The interface is where adoption is won or lost.

08 · DEPLOYMENT

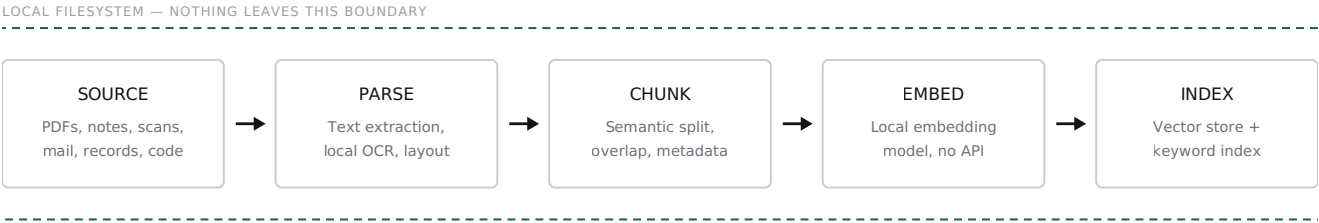
Installation, signed updates that can be applied from media rather than the internet, device attestation, offline licensing, configuration management and rollback. A system that requires a call home to start is not an offline system.

The model is the smallest and most replaceable part of the stack. Design so it can be swapped without touching anything above it.

The data pipeline

Everything the assistant knows about you arrives through this pipeline. Its defining property is that every stage executes on local hardware, including the ones that are conventionally outsourced.

FIG. 8 · Ingestion and indexing — end to end, on local hardware



Re-indexing is incremental: only changed files are re-embedded. Deleting a source deletes its vectors.
No document, embedding or filename is transmitted at any stage.

The embedding step is the one most commonly outsourced to an API by default. Doing so transmits the full text of every document indexed — which reintroduces exactly the exposure the deployment was meant to remove.

6.1 Stage notes

STAGE	WHAT IT DOES	FAILURE MODE TO AVOID
SOURCE	Watches folders, mailboxes, scanners and record systems for new or changed material.	Indexing material the user did not intend to share with colleagues on a shared node.
PARSE	Extracts text and structure; local OCR for scans and photographed documents; preserves tables and headings.	Cloud OCR services — a silent egress of the most sensitive documents in the set.
CHUNK	Splits on semantic boundaries with overlap, carrying metadata: source, page, date, author, access level.	Fixed-size splitting that severs tables and clauses, producing confidently wrong retrieval.
EMBED	Converts chunks to vectors using a local embedding model sized to the hardware.	API embeddings; also, changing embedding model without re-indexing the corpus.
INDEX	Writes to a local vector store plus a keyword index, with per-document access control.	An index that ignores the permissions of the underlying files.

6.2 Deletion actually deletes

Because the index is a local artefact, removing a source document removes its vectors, and removing a user removes their memory and their history. There is no third-party retention window, no training-corpus question, and no need to request deletion from anyone. For GDPR erasure requests and equivalent obligations this converts a legal process into a filesystem operation.

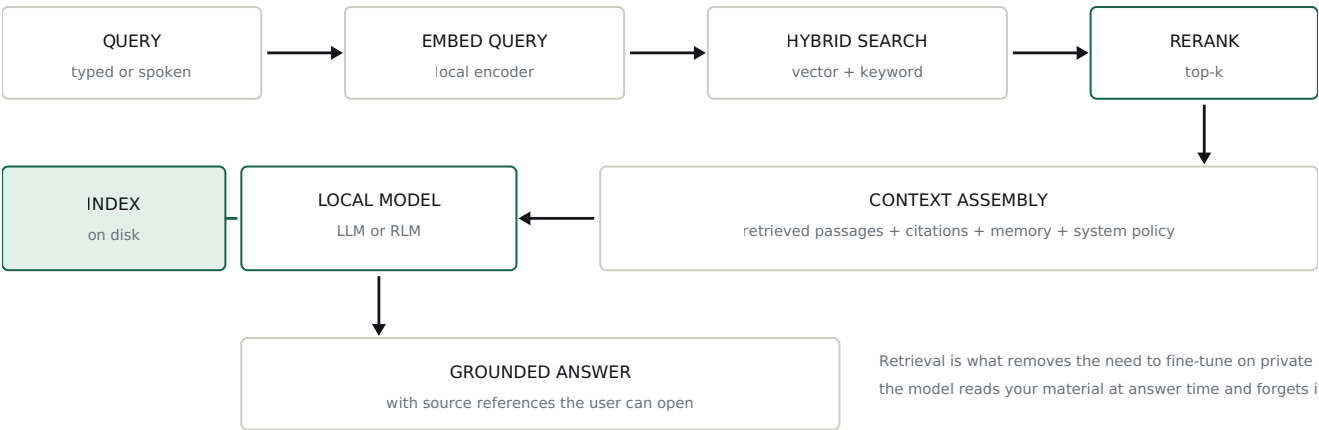
ACCESS CONTROL IS A PIPELINE PROPERTY, NOT A UI FEATURE

On a shared on-prem node, retrieval must respect the permissions of the source system. The index stores an access label per chunk, and the retrieval query is filtered by the requesting user's entitlements before ranking — never after. A retrieval layer that ranks first and filters second will eventually surface a passage from a document the user cannot open.

Retrieval — RAG on private material

Retrieval-augmented generation is the mechanism that makes a modest local model genuinely useful on your material: rather than knowing your documents, it reads the relevant part of them at the moment you ask.

FIG. 9 · The retrieval path



Hybrid search matters on private corpora: pure vector search misses exact identifiers — case numbers, drug names, part codes, function names — that keyword search finds reliably.

7.1 Why retrieval rather than fine-tuning

PROPERTY	FINE-TUNING ON PRIVATE DATA	LOCAL RETRIEVAL
Update latency	Hours to days per training run	Seconds — index the new file
Deletion	Hard; the knowledge is in the weights	Delete the document, delete the vectors
Attribution	None; the model asserts	Every claim carries its source passage
Access control	One model per permission tier	Filter at query time, one model
Cost	GPU hours per iteration	Embedding cost once per document
Leakage risk	Weights can memorise and re-emit	Context is discarded after the answer

Fine-tuning remains the right tool for teaching a model a format, a tone or a domain vocabulary. It is the wrong tool for teaching it facts that change or that someone may later have the right to have erased.

7.2 The quality controls that matter locally

- **Reranking.** First-pass retrieval optimises for recall; a small cross-encoder reranker reorders the shortlist for precision. On a local stack this is usually the single highest-yield addition.
- **Citation by construction.** The interface should show which passage produced which sentence, and let the user open the source. In regulated work this is not a nicety — it is the audit.
- **Refusal on empty retrieval.** If nothing relevant is found, the correct behaviour is to say so, not to answer from parametric memory. This single rule removes most hallucination complaints.
- **Evaluation on your own corpus.** Public benchmarks predict little about performance on one hospital's notes or one firm's contracts. Build a fifty-question gold set from real queries before choosing a model.

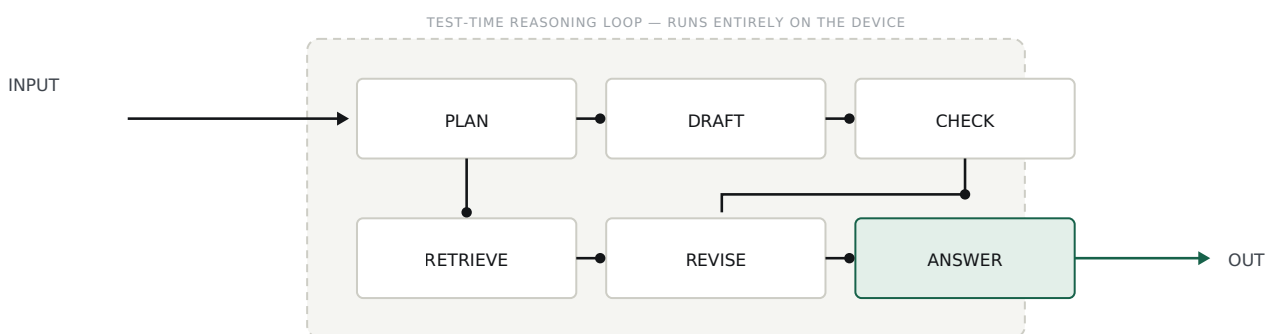
Reasoning models (RLM) at the edge

The development that makes on-premise AI viable in 2026 is not that hardware got cheaper. It is that a class of model emerged which buys accuracy with time rather than with parameters — and time is the one resource a local deployment has in abundance.

8.1 What an RLM is

A reasoning language model spends additional computation at answer time — proposing an approach, drafting, checking its own work, retrieving more evidence, and revising — instead of relying solely on a single forward pass. The practical consequence is that a small model given more time can match a much larger model given less, on exactly the tasks where correctness matters: multi-step reasoning, mathematics, code, structured extraction and cross-referencing.

FIG. 10 · The test-time reasoning loop



A reasoning model spends more compute per question instead of needing more parameters. That trade is what puts genuinely useful reasoning inside a 16 GB laptop — the loop is slower, but it is yours and it is private.

Each pass through the loop is inference on local hardware. The trade is latency for accuracy — acceptable for a research question or a claims review, less so for live dictation.

A cloud deployment pays for reasoning tokens on every single query. An owned machine pays for them in electricity and patience.

8.2 Why this changes the local calculation

- **Parameter count stops being the whole story.** The question shifts from "can I fit a frontier-scale model?" — where the answer is no — to "can I afford to let a small model think for ninety seconds?", where it is usually yes.
- **Quantisation compounds it.** Four- and five-bit quantisation cuts memory requirements several-fold with modest quality loss, putting capable reasoning models inside consumer memory budgets.
- **Batch work suits it perfectly.** Overnight review of a day's documents, weekly literature sweeps and bulk extraction do not care about latency at all.
- **Distillation keeps arriving.** Reasoning traces from large models are used to train small ones, and several such distilled families ship under permissive licences.

8.3 Routing between model classes

TASK	MODEL CLASS	WHY
Dictation, live transcription	Small speech model	Latency dominates; accuracy is adequate
Summarise, rewrite, classify	Small instruct LLM	Single pass is sufficient
Answer over documents	Instruct LLM + retrieval	Knowledge comes from the index, not the weights
Cross-check, reconcile, plan	RLM (reasoning)	Multi-step correctness matters more than speed
Code generation and review	Code-specialised model	Domain-tuned models beat larger generalists here

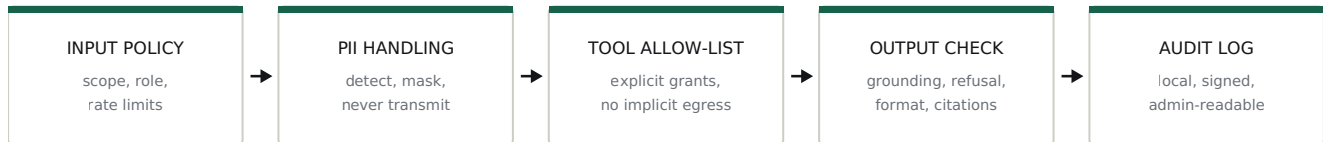
A production stack runs several models and routes between them. Holding two or three quantised models resident is generally cheaper than running one large one.

Guardrails and audit

Running locally removes the transmission risk. It does not remove the behaviour risk — and it transfers responsibility for that behaviour from the provider to you.

FIG. 11 · The policy path around every request

EVERY STAGE EXECUTES LOCALLY — A POLICY THAT DEPENDS ON A REMOTE CALL IS NOT A GUARDRAIL, IT IS A DEPENDENCY



The audit log is the compliance artefact: it proves what the assistant was asked, what it retrieved and what it answered — without any of that evidence being held by a third party.

Guardrails that call a remote moderation service reintroduce both an egress and a dependency. On-premise policy must be enforceable with the network unplugged.

9.1 What each stage enforces

- **Input policy.** Which users may ask what, over which corpora, at what rate. Role-scoped: a receptionist and a consultant should not have the same retrieval reach.
- **PII handling.** Detection and masking where required — and, critically, a hard rule that identifiers are never included in any outbound call, including telemetry and crash reports.
- **Tool allow-list.** The workflow engine may only call tools explicitly granted. Any tool with network reach is separately flagged, and default-denied.
- **Output checks.** Grounding verification against retrieved passages, refusal when retrieval is empty, format validation for structured outputs, and domain rules — a clinical assistant that declines to state a dose, for example.
- **Audit log.** Local, append-only, signed: who asked, what was retrieved, what was answered, which tools ran. This is the artefact an auditor wants, and holding it locally means the evidence of your compliance is not itself in someone else's custody.

HONEST NOTE ON LIABILITY

Self-hosting shifts responsibility. With a cloud provider, model safety behaviour is largely their engineering problem and their published policy. On your own hardware, the guardrail layer is yours to build, test and maintain — and in a regulated setting, yours to evidence. Organisations that treat guardrails as an afterthought generally find their on-prem deployment stalls at the compliance review, not the technical one.

9.2 Where the regulatory advantage is real

Data residency becomes trivially provable: the data is in the building, on hardware with an asset tag. Cross-border transfer questions do not arise. Subject access and erasure requests resolve to local operations. Retention is whatever your own policy says, because no third party holds a copy with a different schedule. For HIPAA, GDPR and comparable frameworks, this collapses a large category of vendor due diligence, processor agreements and transfer-mechanism analysis into a much smaller one.

Memory — the scholar blueprint

Retrieval gives an assistant access to your documents. Memory gives it continuity — knowledge of what you have already covered, what you keep getting wrong, and what you are working towards. It is also the most sensitive artefact the system holds.

10.1 Three kinds of memory

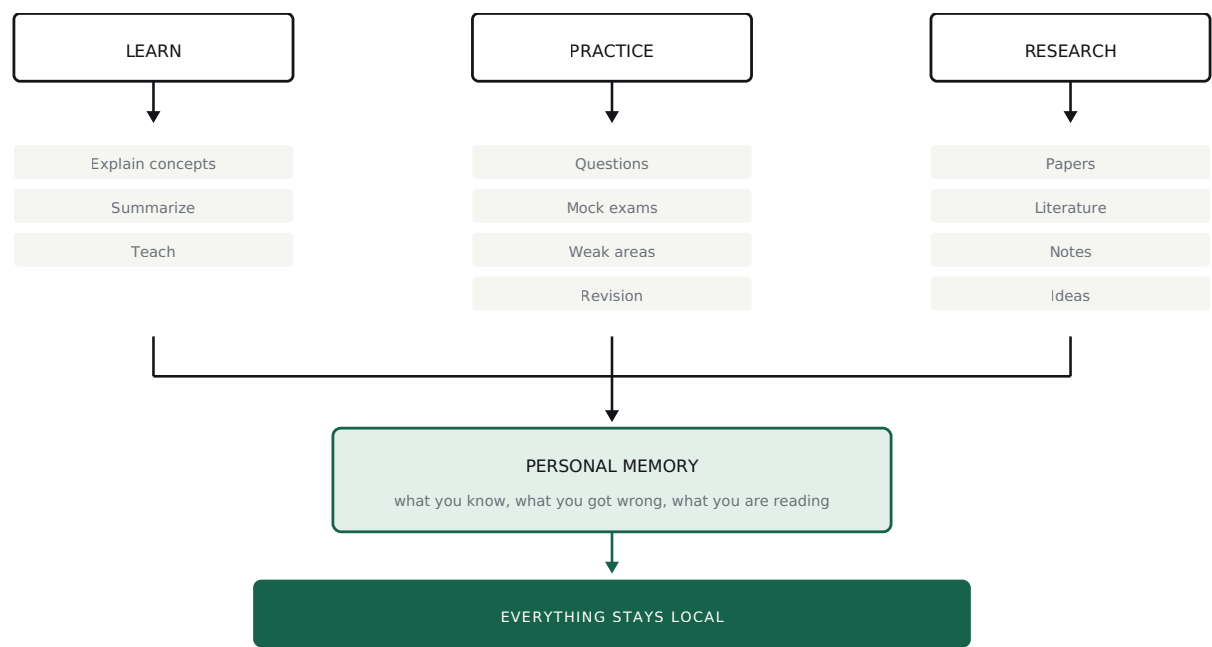
TYPE	HOLDS	LIFETIME
WORKING	The current conversation, the documents in play, the task in progress.	The session
EPISODIC	What was done and when: questions asked, answers given, corrections made.	Until the user clears it
SEMANTIC	Durable facts and preferences: your subject, your syllabus, your writing style, your recurring gaps.	Persistent, editable

All three are ordinary files on the user's disk. They can be read, edited, backed up, moved to a new machine, or deleted — without asking anyone.

10.2 Worked blueprint: the private study assistant

The scholar case shows why memory and locality belong together. A study assistant is only valuable if it knows what you do not know — which means it necessarily accumulates a detailed, unflattering, longitudinal record of your weaknesses. That record should not be a line item in an external retention schedule.

FIG. 12 · Three modes, one memory, no egress



Each mode writes to the same personal memory, and each reads from it: practice targets the gaps that learning revealed, and research is filtered by what you already know.

WHAT EACH MODE CONTRIBUTES

- **Learn** — explains concepts at your level, summarises material, and teaches from your own notes and set texts rather than from generic web content.

- **Practice** — generates questions and mock exams from the syllabus, marks them, tracks which topics fail repeatedly, and schedules revision against your exam date.
- **Research** — reads papers and long documents, maintains a literature map, links new reading to existing notes, and surfaces contradictions with what you have already recorded.

The same pattern generalises directly: a clinician's memory is their patient cohort and their protocol preferences; an engineer's is their codebase conventions and past incidents; a firm's is its precedent and house style. In every case the value of the memory rises with its specificity, and so does the harm of losing control of it.

An assistant worth having knows things about you that you would not put in a form. That is precisely why it should live on your side of the wall.

The open-source foundation

On-premise AI is practical today because the components are permissively licensed. MIT and Apache-2.0 licensing is what makes it legally possible to run a model on your own hardware, indefinitely, without a per-seat agreement or a phone-home check.

11.1 Why the licence class is the first question

Before capability, before benchmarks: a model you cannot legally deploy in your jurisdiction, at your scale, for your purpose, is not a candidate. Three classes matter.

CLASS	WHAT IT PERMITS	WATCH FOR
MIT	Commercial use, modification, redistribution, private deployment. Minimal text, minimal friction.	No explicit patent grant.
Apache 2.0	The same freedoms, plus an explicit patent grant — which enterprise counsel often prefer.	Attribution and notice requirements.
Community / bespoke	Weights download freely, but the licence restricts use.	Scale thresholds, field-of-use limits and territorial restrictions that have in the past excluded EU entities.

11.2 The current landscape

As of mid-2026 the permissively licensed field is broad. Several major open-weight families ship under MIT — including the DeepSeek line and Microsoft's Phi models — while Qwen and Mistral's open tiers ship under Apache 2.0. Meta's Llama family remains under a community licence with scale and territorial conditions that must be read before deployment.

VERIFY AT ADOPTION, NOT FROM THIS TABLE

Model licences change between releases, and different sizes within one family sometimes carry different terms. Treat any licence summary — including this one — as a starting point, and confirm against the model card and licence file for the exact weights you intend to ship. Where a deployment is regulated, have counsel review the licence text itself.

11.3 A representative permissive stack

The layers of Section 5 can be assembled almost entirely from permissively licensed projects. The following is illustrative of what a build can be composed from, not a product specification.

LAYER	COMPONENT CLASS	TYPICAL LICENCE CLASS
MODEL RUNTIME	Local inference engines and model servers	MIT / Apache 2.0
WEIGHTS	Open-weight instruct and reasoning models	MIT / Apache 2.0
EMBEDDINGS	Local sentence and document encoders	Apache 2.0
VECTOR INDEX	Similarity search libraries and vector databases	MIT / Apache 2.0
KEYWORD INDEX	Full-text search engines	Apache 2.0 / public domain
ORCHESTRATION	Retrieval and agent frameworks	MIT
SPEECH	Local transcription models	MIT
STORAGE	Embedded databases for memory and audit	Public domain / Apache 2.0

Licence classes are given at component-class level deliberately. Specific projects change licences — several notable AI tools have relicensed since 2024 — so the obligation to check falls at the point of adoption.

11.4 What permissive licensing buys the customer

- **No expiry.** Weights on disk keep working when a vendor discontinues a product line.
- **No metering.** Cost is hardware and power, not usage.
- **Auditability.** Source can be read, patched and pinned — a requirement in some public-sector procurement.
- **Exit.** Nothing about the deployment is contingent on a commercial relationship continuing.

Hardware sizing and deployment tiers

The single most common cause of a disappointing on-prem pilot is a mismatch between the model chosen and the memory available. Size the hardware to the job, then choose the largest model that fits comfortably — not the largest that fits at all.

12.1 Indicative tiers

TIER	TYPICAL MEMORY	REALISTIC WORKLOAD	SERVES
DEVICE	8–16 GB unified	Summarisation, drafting, classification, transcription, retrieval over a personal corpus. Small reasoning models with patience.	One user, offline
WORKSTATION	24–48 GB VRAM	Mid-size instruct and reasoning models at conversational speed; document-scale retrieval; code assistance.	1–10 users
ON-PREM NODE	48–96 GB VRAM	Large open-weight models, concurrent sessions, batch pipelines, multiple resident models with routing.	10–100 users
PRIVATE RACK	Multi-GPU	Frontier-scale open weights, heavy concurrency, fine-tuning, high-throughput batch reasoning.	Whole organisation

Memory figures assume 4–5 bit quantised weights plus KV cache headroom for the intended context length. Long-context work raises the requirement substantially — context is often the binding constraint, not parameter count.

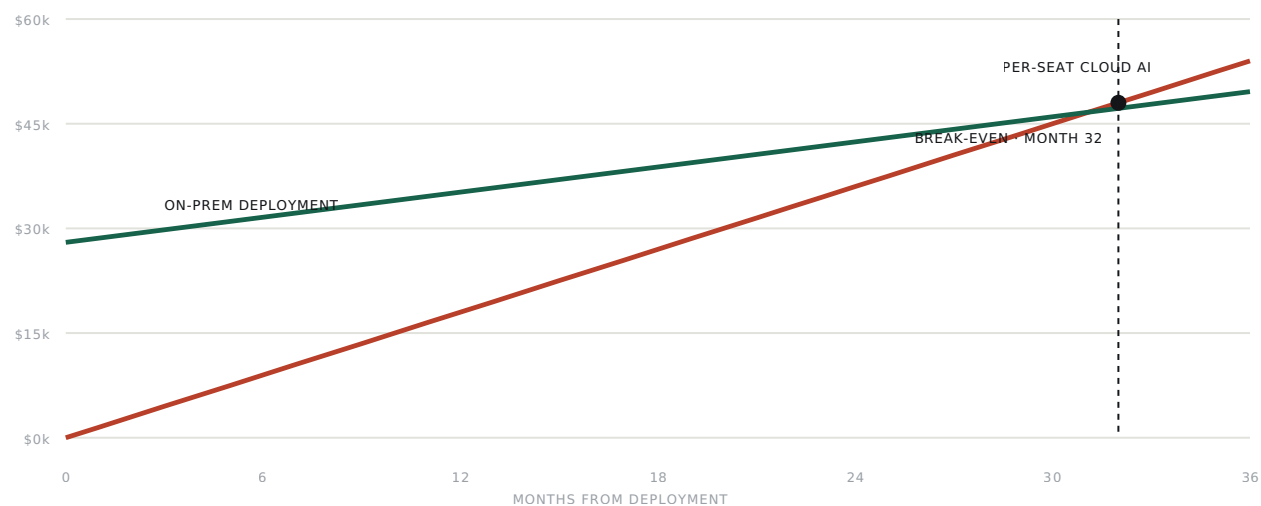
12.2 Sizing rules of thumb

- **Budget roughly the quantised weight size plus 30–50% headroom** for cache and runtime.
- **Prefer a smaller model with a longer usable context** over a larger one that forces aggressive truncation — retrieval quality usually beats raw parameter count on private corpora.
- **Separate interactive from batch.** Overnight reasoning jobs can use hardware that would be unacceptably slow for chat.
- **Plan for two or three resident models** rather than one, and route by task class.
- **Measure on your own corpus before purchasing.** A one-week pilot on a rented workstation is cheaper than a mis-specified capital purchase.

Economics and break-even

Cost is the weakest of the arguments for on-premise AI, and it is worth being straightforward about that. Below a certain scale, subscriptions win. The following model shows where the line falls and which assumptions move it.

FIG. 13 · Cumulative 36-month cost — illustrative model



Assumptions: 50 knowledge workers; per-seat AI subscription at \$30/user/month; one on-prem inference node at \$28,000 capital cost; \$600/month for power, maintenance and administration. Hardware is treated as a single upfront cost with no residual value. This is a model for reasoning about shape, not a quotation.

WHAT MOVES THE CROSSOVER

FACTOR	EFFECT ON BREAK-EVEN
More seats	Earlier — subscription cost scales with headcount, the node largely does not
Heavy per-user usage	Earlier — metered plans grow with tokens; owned hardware does not
Larger model ambition	Later — GPU capital cost rises steeply with memory
Existing infrastructure and staff	Earlier — marginal operational cost is much lower
Small team, occasional use	May never break even — subscriptions are the rational choice

COSTS THIS MODEL DELIBERATELY EXCLUDES

Integration engineering, evaluation work, guardrail development and ongoing model updates are real and non-trivial, and they fall on the customer in an on-prem deployment. Equally excluded on the other side: vendor due diligence, processor agreements, transfer-mechanism analysis, and the expected cost of a breach — which, per Section 2, averages \$4.44M globally and \$7.42M in healthcare. A serious business case should include all of these rather than the ones that favour the preferred answer.

If the only reason to run locally is cost, run the numbers honestly and be prepared for the answer to be no. If the reason is custody, cost is not the deciding term.

Sector blueprints

The same stack, configured four ways. Each blueprint states what runs locally, what the memory holds, and what the guardrail layer must enforce.

Healthcare

Deployment: on-prem node inside the clinical network, no external route. **Corpus:** protocols, formulary, guidelines, de-identified notes where policy permits. **Memory:** clinician preferences, department protocols, prior queries. **Guardrails:** role-scoped retrieval, no dosing or diagnostic assertion without citation, mandatory human confirmation on any clinical output, full audit log retained locally.

Context: healthcare breaches averaged \$7.42M in 2025 and took around 279 days to identify and contain — the longest of any sector [1][5].

Scholars and research

Deployment: workstation or laptop, fully offline capable. **Corpus:** papers, lecture notes, textbooks, personal writing, unpublished results. **Memory:** the model of Section 10 — syllabus, mastery, weak areas, reading history. **Guardrails:** citation required for factual claims; refusal when retrieval is empty; a hard block on any egress of unpublished work.

Home

Deployment: a single always-on device on the home network. **Corpus:** household documents, manuals, personal records. **Memory:** routines, preferences, household context. **Guardrails:** no audio or video leaves the property; automation actions require confirmation; guests get a scoped profile. The dependability argument is sharpest here — a home assistant that stops working during a provider incident was never really in the home.

Office and enterprise

Deployment: on-prem node or private rack, integrated with identity. **Corpus:** internal knowledge, contracts, tickets, meeting records, code. **Memory:** team conventions, project state, decision history. **Guardrails:** permissions inherited from source systems, default-closed tool access, retention aligned to existing records policy, exportable audit for compliance review.

Context: organisations with high shadow-AI use paid roughly \$670,000 more per breach; a sanctioned internal assistant is, among other things, the most effective shadow-AI control available [1].

Honest limitations

On-premise AI is the right answer for a large and growing set of problems. It is not the right answer for all of them, and a paper that claimed otherwise would not be worth reading.

WHERE THE CLOUD REMAINS BETTER

- **Absolute capability ceiling.** The largest frontier models are not runnable on a workstation, and for the hardest reasoning, longest contexts and richest multimodal work they remain ahead of anything you can self-host.
- **Zero operational burden.** An API has no hardware to buy, no updates to schedule and no capacity to plan. That is a genuine and permanent advantage.
- **Elastic bursts.** Occasional very heavy workloads are cheaper rented than owned.
- **Small teams.** Below roughly ten regular users, the economics rarely favour ownership.

WHAT YOU TAKE ON

- **You own the safety layer.** Guardrails, evaluation and their maintenance become internal engineering work.
- **You own the update cadence.** Stability is a benefit; falling three model generations behind is not.
- **You own the hardware lifecycle.** Capacity planning, failure, refresh and disposal.
- **Local is not automatically secure.** An on-prem node with weak access control, an unfiltered index and no audit trail can leak internally just as effectively as an external service leaks outward. Locality removes one class of risk; it does not remove the discipline.

A TEST FOR WHETHER THIS APPLIES TO YOU

Ask what happens if the material in question appears in a breach notification, a discovery request or a training corpus. If the answer is "a difficult conversation," subscriptions are fine. If the answer is "regulatory action, patient harm, or the loss of unpublished work," the custody question is the deciding one and cost is secondary.

A 90-day implementation path

The failure pattern for on-prem AI projects is to begin with a hardware purchase. Begin instead with a measured question set, and let it choose the hardware.

PHASE	OBJECTIVE	WORK
DAYS 1-15	Define the job	Pick one workflow with real value and real sensitivity. Write 50 questions from actual use, with correct answers. This gold set is the project's only objective measure.
DAYS 16-30	Prove the pipeline	Index a representative corpus locally. Test retrieval quality alone — no generation. Most quality problems are retrieval problems and are cheapest to fix here.
DAYS 31-50	Choose the model	Run two or three permissively licensed candidates against the gold set on rented or borrowed hardware. Record accuracy, latency and memory. Choose on evidence, not leaderboard.
DAYS 51-70	Build the policy layer	Guardrails, access control inherited from source systems, audit logging. Review with whoever signs off compliance, before scaling.
DAYS 71-90	Pilot with real users	Ten users, daily use, structured feedback. Measure adoption, not satisfaction. Then size and buy hardware against observed load.

SIGNALS TO STOP

- Retrieval cannot reach 80% on the gold set — fix the corpus and chunking before blaming the model.
- Pilot users revert to the cloud tool within two weeks — the interface or the latency is wrong.
- Compliance cannot sign off the audit trail — that gap will not close later.

Ship one workflow that people use daily. A stack that does ten things adequately loses to a cloud tab that does one thing well.

References

1. IBM Security & Ponemon Institute. *Cost of a Data Breach Report 2025*. 600 organisations, 17 industries, 16 countries, breaches occurring March 2024 – February 2025. ibm.com/reports/data-breach
2. IBM Think. *2025 Cost of a Data Breach: navigating the AI rush without sidelining security*. 2025.
3. LayerX. *Enterprise AI and SaaS Data Security Report*. 2025.
4. Cyberhaven. Research on corporate data exposure through generative AI tools. 2025.
5. HIPAA Journal. *Average cost of a healthcare data breach*. 2025.
6. Cisco ThousandEyes. *AWS outage analysis, 20 October 2025*. thousandeyes.com
7. Amazon Web Services post-event summary, October 2025; aggregate public outage reporting.
8. Comparative edge-versus-cloud inference performance literature, 2024–2025.
9. *Edge-first language model inference: models, metrics and tradeoffs*. arXiv:2505.16508.
10. Stanford HAI. *AI Index Report 2025* — AI-related security incident tracking.
11. Published open-weight model licence terms and model cards, as surveyed mid-2026.

SECTION 18 · LEGAL NOTICE

Status. No IOT is a pre-release company. This document describes an architectural approach and the published state of the field. It is not a product specification, a performance benchmark, or a claim about any shipping No IOT product.

Third-party figures. All statistics are attributed to their published sources and describe the wider industry. They are reproduced in summary form; readers relying on them for decisions should consult the original reports, which may have been updated since publication of this paper.

Licensing. References to MIT, Apache-2.0 and other licences are general descriptions current at the time of writing. Licence terms vary between projects, between model families and between releases within a family. Verify the licence file and model card for the exact components you deploy, and obtain legal advice for regulated deployments.

Not professional advice. Nothing here constitutes legal, regulatory, clinical or financial advice. Compliance obligations under HIPAA, GDPR and comparable frameworks depend on specifics this document cannot assess.

© 2026 No IOT. All rights reserved. This white paper may be shared unmodified with attribution. www.noiot.ai

— END —