

Domain-specific AI on your own hardware

*Capability, cost, and the case
for institutional ownership*

Chris Kellett, GAICD, EngExec, CPEng, FIEAust, FAustMS, FIEEE
Chief Research Officer, RACE for 2030 CRC
Principal Researcher, ResTech Pty Ltd
Honorary Professor, University of Newcastle

September 2026

Acknowledgement of Country

The author of this report respectfully acknowledges the Traditional Owners of the ancestral lands throughout Australia and their connection to land, sea and community. He recognises their continuing connection to the land, waters, and culture and pays his respects to them, their cultures and to their Elders past and present.

Copyright

© ResTech Pty Ltd 2026

This report is licensed under a Creative Commons Attribution 4.0 International Licence (CC BY 4.0). You are free to copy, distribute, adapt and build upon this work, including commercially, provided you attribute it appropriately, indicate if changes were made, and do not suggest that ResTech Pty Ltd or RACE for 2030 endorses you or your use. The full licence terms are available at creativecommons.org/licenses/by/4.0/.

This licence applies to the content of this report with the following exceptions: the ResTech and RACE for 2030 logos and trade marks; and any third-party material identified as such. Figure 3 is reproduced from Epoch AI under a Creative Commons Attribution licence and remains subject to its own terms and attribution requirements.

Citation

Kellett, C. (2026). *Domain-specific AI on your own hardware: capability, cost, and the case for institutional ownership*. ResTech Pty Ltd, Newcastle

Disclaimer

The author has used all due care and skill to ensure the material is accurate as at the date of this report. The author does not accept any responsibility for any loss that may arise by anyone relying upon its contents.

Interests

ResTech's work for industry clients includes applications adjacent to those discussed in this report, and the author has a material professional interest in this capability developing in Australian industry. The hardware, software, and methodology described are commercially available from multiple vendors, and none involve a financial relationship with the author or his affiliated organisations.

Audience

This document is written for people who have to make institutional decisions about AI deployment and use: leadership at Australian distribution and transmission network service providers, state and federal government stakeholders, and executive leadership of Australian professional bodies. It also addresses research leadership — Vice-Chancellors, Deputy Vice-Chancellors Research and Academic, CSIRO, CRC CEOs and Chief Research Officers, and the federal research funding agencies (ARC, NHMRC, ARENA).

Executive summary

The second AI question

Australian energy institutions are already deep in one conversation about AI and the grid. AEMO's 2026 Electricity Statement of Opportunities counts 225 known data centre projects representing 67 GW of proposed connection capacity. NSW has moved to let the state energy minister authorise or prohibit network access for loads of 5 MW or more. Rule change requests on data centres paying the network costs they cause go to energy ministers in September 2026. That conversation treats AI as a load.

A second question sits alongside it and has attracted much less attention: what AI capability an energy institution should build and own, as against buy as a service. Over the past eighteen months the terms of that question have changed. Hardware that runs and specialises large models now costs A\$7,000–10,000. The models are free. The software is free. What was once a capital decision requiring board attention now sits inside the discretionary spend of a single business unit.

This document sets out what changed, what it costs, what happened when the author built such a system himself, and why the institutions that already certify professional competence are the natural owners of the resulting capability. Its default reader works in a distribution or transmission business, at AEMO, the AEMC, or the AER, or in a state or federal energy department. It also addresses university research leadership, professional body executives, and other regulated professions, because the architecture is identical across all of them and only the corpus changes. A law society curating case law and practice directions, and a distribution business curating its network standards, planning instruments, and connection guidelines, are doing the same work on different material.

What has changed

Hardware, open models, and tooling converged in late 2025, and each has held or improved since.

Hardware. NVIDIA's DGX Spark (~A\$7,200) and the Dell Pro Max with GB10 (~A\$9,400) put 128 GB of unified memory on a desktop, rated for inference on models up to 200 billion parameters at 4-bit quantisation and comfortable fine-tuning to around 30 billion. A DGX Station at A\$130,000–150,000 runs inference on trillion-parameter models. On NVIDIA's published roadmap, current Station capability reaches the Spark price point in about three to four years.

Models. Open-weight models reach approximately 90% of frontier performance at release. Their lag behind the best closed models narrowed sharply and then stabilised: Epoch AI puts the average at about three months to October 2025 and about four months since January 2026. The gap has held at three to four months rather than continuing to close, which is important for an institution planning a deployment on a two-year horizon.

Cost. Over June–July 2026, open-weight models run through commercial providers cost roughly ten to fifty times less per token than proprietary frontier models on OpenRouter, which hosts both. The gap is structural: any provider can host an open-weight model, so prices compete down toward the cost of compute, while a proprietary model is hosted only by the company that made it. On owned hardware the per-token comparison does not apply — after the capital outlay, a query costs electricity.

Tooling. Specialising a model on an organisation's own corpus costs A\$20–50 per run in cloud compute at full commercial rates, or nothing beyond electricity on owned hardware. The open-source toolchains that do it are documented and production-ready, and carry no licence cost.

Domain-specific AI capability — a model that knows a particular institution's technical and regulatory material, running where that material never leaves the building — is therefore available for the price of the hardware and a few weeks of a senior expert's time.

What happened when it was tried

Part 4 tests the practitioner claim directly: whether a senior expert whose hands-on technical work has thinned with seniority can build this himself. The author did, over two weeks in April and May 2026, on the hardware described above, fitted around a full-time job — hardware setup, corpus preparation, training runs, evaluation, and a controlled comparison against two frontier services. The corpus was 800,000 words of his own technical writing, chosen because he could judge every answer it produced. Designs, scoring rubrics, and the mapping from outcomes to conclusions were fixed before any data was visible, so the exercise could fail honestly.

The first approach failed, and failed invisibly. Training the model on the corpus using the default settings from the most accessible public tutorials — what a capable practitioner would reasonably try first — produced a model that wrote fluent, correctly-notated, confidently wrong technical content while scoring normally on standard benchmarks. Someone running those benchmarks would have concluded it worked. The route that did work was retrieval over the curated corpus with the model left untrained. The heavier training routes the literature supports were not carried through to a working build and remain open.

Model choice decided the outcome at a fixed price.

Two models of the same size, one generation apart, on the same machine over the same corpus: the older produced a well-structured answer that was wrong in four places, including a claim the retrieved documents themselves contradict; the newer was scored competitive against Claude Opus 4.7 and GPT-5 on every question — the rubric's band for a difference of no more than three points against the frontier configuration. Of three implementation points tested at this hardware tier, two cleared the threshold set in advance and one did not. An institution cannot pick a model off a specification sheet and skip the evaluation.

The failure mode that matters in a regulated setting did not appear.

Retrieval that matches on vocabulary without matching on substance can draw a confident answer out of the model's general training, cited as though the retrieved documents supported it — a wrong-jurisdiction answer on grid code parameters, carrying grounded-looking citations. Six samples across five configurations were run against a question engineered to bait that behaviour. All six identified the coverage gap and declined or hedged, including the configuration that fabricated elsewhere.

The evidence base is narrower than the argument it supports. A single-author corpus with consistent notation, in a domain the author knows deeply, is a favourable case, and one practitioner is one practitioner. Six samples on one engineered probe show that the architecture can refuse correctly; they do not show that it will over multi-author institutional corpora — RACE reports, AEMO documents, sector publications — where mismatched retrieval is routine rather than staged. That remains untested. The institutional argument below rests on this demonstration together with the surrounding public evidence.

The case for institutional ownership

The engineering friction — terminals, dependency management, breaking changes in fast-moving toolchains — is real today and is being automated away; a web interface that takes a corpus and returns a specialised model is within technical reach, held up by whether anyone sees a market for it. What does not automate is deciding what belongs in the corpus, writing evaluation questions subtle enough to separate a model that has learned the domain from one producing plausible wrong answers, and recognising a wrong answer that reads as competent work. That is the same judgement the tool is being built to support, it decides whether the tool works, and it has to be maintained by someone for as long as the tool is in use.

No single build produces a finished system: what works is a release-and-retune cycle, deploying with expert review in place, collecting failure signals from real practitioner use, and retuning on a sustainable cadence. And a cleanly curated, internally consistent corpus outperforms a larger, messier one, which is what makes curation a domain-judgement task rather than a data-engineering one.

Individual practitioners cannot maintain corpus quality or sustain that cycle across years. Commercial vendors can, but their incentives run toward generality, and the failure signal their users generate accrues to the vendor. Professional bodies and sector institutions are positioned differently: they already curate authoritative domain material — guidelines, practice notes, standards, accreditation frameworks — on annual cycles, they have the continuity to own an asset across decades, and their members generate exactly the usage patterns that reveal where the system is failing.

Continued pre-training is out of reach on desktop hardware at any model size. A practitioner on A\$7,000–10,000 of desktop hardware can run retrieval over a curated corpus, and can fine-tune models up to about 30 billion parameters, but continued pre-training is out of reach at that tier at any model size. An institution is not under that constraint: it can train once on hardware sized for the job, or in the cloud, and distribute the result as adapter weights of a few hundred megabytes that members load alongside the same base model on machines they already own. Where the body instead maintains the corpus, the retrieval index, and the evaluation instrument, members run the whole stack locally and the same asymmetry holds — the curation and evaluation that determine the outcome are beyond what a member sustains alone. Either way the marginal cost of keeping current is small. When a new model was released four months after the work reported in Part 4, testing it took about ninety minutes against the existing corpus, index, rubric, and scoring procedure — most of that spent downloading the model.

What is missing from the conversation

The Australian public conversation about AI has coalesced around four positions: governance and ethics, industry adoption and ecosystem building, infrastructure investment, and academic AI research. Each has institutions, funding, and advocates behind it. Together they leave out a fifth, which is practitioner capability — domain experts and the institutions that certify them building and maintaining domain-specific AI themselves, over the data and standards that are invisible to commercial models. This differs from adoption support, which is about helping practitioners use commercial tools well. The author could not identify an Australian professional body that has publicly committed to building this layer, or a prominent Australian voice making the argument.

The gap has a specific cost in this sector. The National Electricity Rules, AEMO procedures, AER guidelines, the AS/NZS standards series, and state planning instruments are a statistical rounding error in the global English corpus that every frontier model is trained on. A model that sounds Australian and does not know the National Electricity Rules is useful for general tasks and useless to the analyst who needs it. Nobody outside the sector is going to curate that material.

There is a second-order consequence for the energy system itself. The dominant policy framing treats AI compute as a hyperscaler workload requiring transmission-connected capacity and new substations. The Spark described in Part 4 has run for months without tripping a circuit breaker or producing a noticeable change in household electricity use. A DGX Station is rated at 3 to 5 kW. Distributing part of the sector's own AI workload across desktop devices on the existing distribution network is a different energy proposition from concentrating it in a handful of facilities. This does not dissolve the data centre question — some tasks require frontier compute, some of the projected demand is regional compute export rather than domestic AI, and data centres carry far more than AI workloads. It adds a variable the connection-queue framing does not currently contain.

This document does not argue that every institution should build this capability. It argues that the decision has become a real one, at a price point that no longer forces it upward through the organisation, and that the institutions best placed to make it are not currently being asked to.



A\$7,000–10,000 desktop hardware



A\$20–50 per fine-tuning run
(cloud, full commercial rate) — or
electricity on owned hardware



~90% of frontier performance at release



3–4 months open/closed capability gap,
stable since early 2026



Zero ongoing software cost

Part 1: what has changed

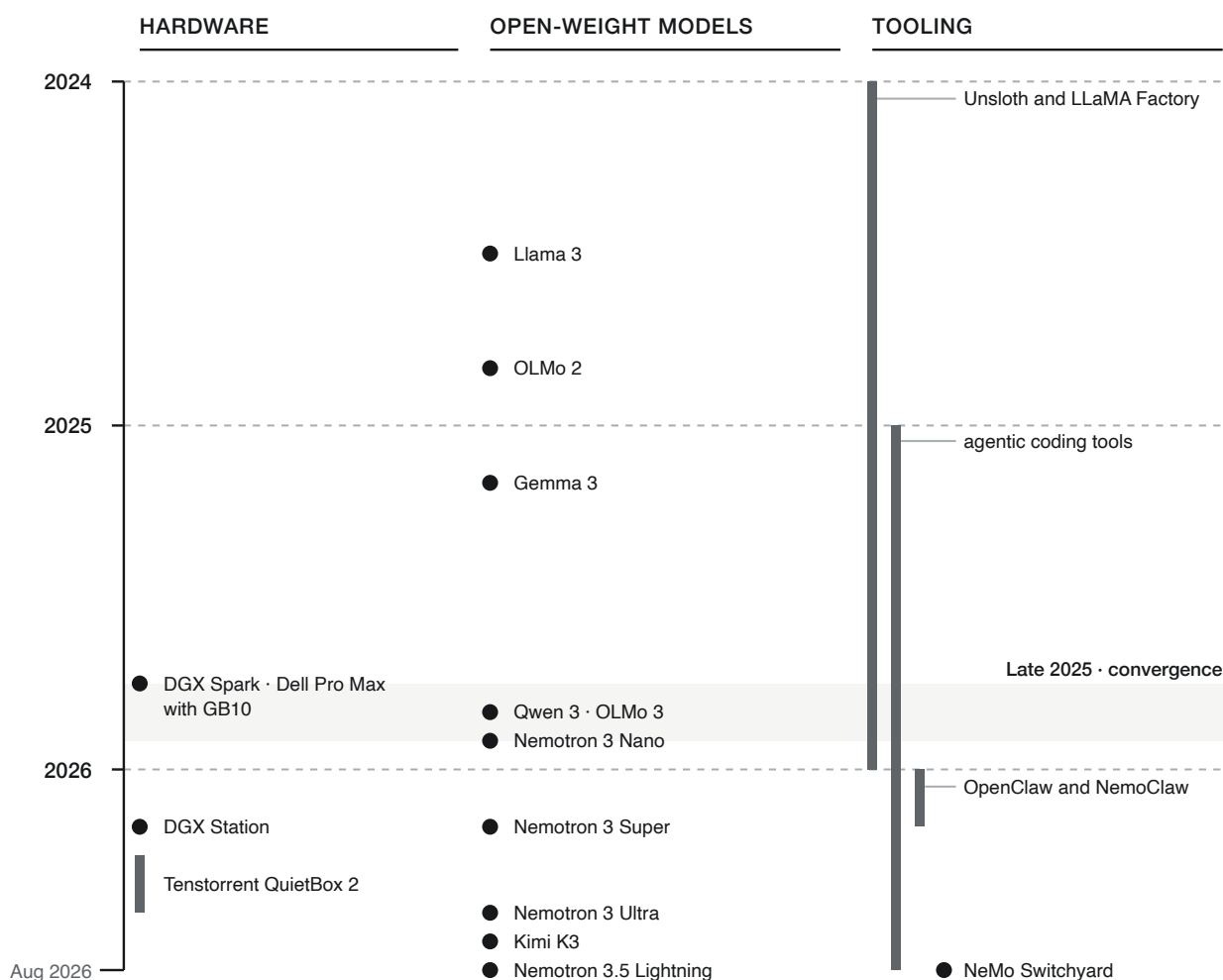
1.1 The convergence

Most of what this report describes did not exist eighteen months ago. Some of it did not exist six months ago. The convergence of hardware, algorithms, and accessible (though not yet frictionless) software tools has been rapid, and it is still accelerating.

NVIDIA's DGX Spark — a desktop machine in the A\$7,000–10,000 range capable of running and fine-tuning AI models that were datacentre-only hardware two years ago — started shipping in October 2025. Comparable non-NVIDIA options, such as Tenstorrent's QuietBox 2, started shipping early this year. NVIDIA's DGX Station, which runs

models up to one trillion parameters on a desktop, started shipping in March 2026. The open-source agentic coding tools that allow a domain expert to direct AI through complex research and development workflows are 2025–2026 products. The open-weight AI models that run on this hardware have closed the performance gap with proprietary frontier models from a year to approximately three months, with that gap continuing to narrow. The genuinely open-source foundation model OLMo 3, which can be fully audited and reproduced, was released in November 2025. NVIDIA's Nemotron 3 family, released across late 2025 to early 2026 with open weights, training data, and recipes, is specifically optimised for NVIDIA's own desktop hardware.

Dots mark a release date; bars mark a period.



The three tracks converge in late 2025, when hardware, open-weight models and tooling become simultaneously available to a domain expert at desktop scale for the first time. Models accelerate through 2026; hardware continues at a steady cadence.

Figure 1. The convergence, 2024–2026. Hardware, open-weight models and tooling became simultaneously available to a domain expert at desktop scale in late 2025. The model track has accelerated since; the hardware track has held steady.

This pace creates a problem for decision-makers. The public discourse about AI is dominated by consumer novelty, speculative futures, and conference demonstrations of eye-catching but peripheral applications. This leaves people with no framework for understanding what is practically possible right now, at what cost, and with what implications for how their organisations operate.

Three core developments:

Affordable local hardware. Desktop AI systems that run models previously requiring datacentre infrastructure, at price points from ~A\$7,000 to ~A\$150,000, with a clear roadmap forecasting 3x improvement per generation.

Competitive open-weight models. AI models with publicly available weights now reach approximately 90% of frontier model performance at release and close the remaining gap within a few months. Running these models through commercial providers was an order of magnitude cheaper than running proprietary frontier models on OpenRouter in June 2026 (see Section 1.3). A genuinely open-source alternative exists for organisations that need full auditability.

Mature tooling for both fine-tuning and interactive use. The software to specialise models on domain-specific data is production-ready. Separately, AI-powered development tools — both proprietary and open-source — now enable expert-directed research, coding, simulation, and analysis at dramatically compressed timescales. Agentic AI tools that can autonomously execute multi-step workflows are emerging.

The combined effect is that a small organisation with domain expertise can now build and deploy AI capability that, on domain-specific tasks, matches or exceeds what the largest proprietary models offer — and can use AI as an interactive R&D partner to accelerate the full cycle from question to analysis to output. This is happening now, with commercially available hardware, using free software, at price points accessible to any professional firm or research group.

The dominant AI conversation over the past six months has been about agentic AI — systems that take autonomous actions rather than responding to individual queries. This report primarily addresses the prior question: whether the model's domain knowledge and reasoning are trustworthy enough to act on. Agency without reliable domain reasoning risks amplifying errors.

What AI means in this doc

The word “AI” is used throughout this report as shorthand for one specific branch of a much broader field. This sidebar establishes the scope and acknowledges what the scope leaves out.

Artificial intelligence as a discipline encompasses many distinct technologies developed over roughly seven decades: classical machine learning (regression, decision trees, support vector machines, ensemble methods), computer vision, reinforcement learning, robotics, expert systems, symbolic reasoning, planning, and natural language processing. Each has its own research traditions, its own application domains, and its own distinct relationship with questions of capability, deployment, and institutional ownership.

The substantial shift this document describes — desktop-accessible hardware, open-weight models, mature fine-tuning tooling, commodified deployment — is a shift that has happened over the past eighteen months for large language models and their associated tooling. It has not happened to the same degree, or on the same timeline, for computer vision, reinforcement learning, robotics, or most other branches of AI.

Throughout this document, “AI” refers to large language models (the technology behind ChatGPT, Claude, Gemini, OLMo, and Nemotron) and the tools that use them as their underlying engine — retrieval-augmented generation systems, agentic coding tools, interactive research assistants, and the fine-tuning infrastructure that specialises these models for particular domains. Where this report discusses “AI tools” or “AI capability,” it means this specific cluster of technologies, not the broader field.

The structural arguments the document develops — about practitioner capability, professional body ownership, domain-specific fine-tuning, the temporal mismatch between institutional research and deployable capability — may generalise to other AI disciplines as their capability profile shifts in comparable ways. Classical machine learning has been desktop-accessible for decades, but as a technical optimisation tool used by specialists. It has not attracted the institutional attention large language models have because its outputs are narrower, its interface more technical, and its scope task-specific rather than conversational. What makes language models distinctive in August 2026 is not that they run on a desktop, but that a desktop-accessible capability is now general enough, interactive enough, and substantive enough to be worth a domain expert's time across a broad range of tasks — and that the tooling around it has matured to the point where specialising the capability for a particular domain no longer requires a machine learning specialist. The arguments in this document are specific to that configuration, not to language models as a technology. Other AI disciplines may reach comparable configurations on their own timelines, at which point the structural arguments would apply to them too.

1.2 Hardware

Desktop and local

Prices quoted in this report are in Australian dollars, current as of August 2026, and subject to currency movement and supplier variation. Specific figures are given where useful. The A\$7,000–10,000 and A\$130,000–150,000 ranges used elsewhere in this report cover routine variation with headroom.

NVIDIA DGX Spark (~A\$7,200). Shipping since October 2025. Also available as the Dell Pro Max with GB10 (~A\$9,400) through Dell's standard commercial channel — same GB10 hardware and DGX OS, orderable through existing Dell procurement accounts. The first generation of NVIDIA's desktop AI category. Built around the GB10 Grace Blackwell Superchip with 128 GB of unified CPU-GPU memory. NVIDIA rates the Spark for inference on models up to 200 billion parameters (at 4-bit quantisation) and fine-tuning up to 70 billion parameters (using QLoRA). In practice, independent reviewers report the comfortable fine-tuning range is up to around 30 billion parameters, with larger models requiring significantly more time and encountering thermal management constraints. This is first-generation hardware. NVIDIA's published roadmap has the Spark line moving to the Vera Rubin architecture in 2027–28, so these practical limits should shift substantially with the next generation. Compact desktop form factor (roughly 15 cm square). Preloaded with NVIDIA's full AI software stack. Two units can be clustered via high-speed interconnect for a combined 256 GB memory pool.

NVIDIA DGX Station (~A\$130,000–150,000). Started shipping March 2026 from OEM partners (ASUS, Dell, MSI, Supermicro). Built around the GB300 Grace Blackwell Ultra Desktop Superchip with 748 GB of coherent memory — 252 GB of high-bandwidth GPU memory plus 496 GB of slower system memory — and 20 petaflops of AI compute. Runs inference on models up to one trillion parameters on a desktop, comparable to last-generation frontier models. The current frontier models likely sit around 2–5 trillion parameters (details are unpublished) and remain beyond desktop hardware. The gap is roughly one generation. The current hardware trajectory, coupled with ongoing architectural research leading to compute efficiency gains, suggests the gap will close or at least not widen. Supports air-gapped configurations where no data leaves the machine. NVIDIA rates the Station for fine-tuning models up to one trillion parameters; third-party assessments put the practical range up to about 400 billion.

Tenstorrent TT-QuietBox 2 (~A\$15,500). Announced March 2026, shipping from Q2. Four Blackhole RISC-V accelerators with 128 GB of GDDR6 plus 256 GB of system memory, with vendor claims of running models up to 120 billion parameters locally (Tenstorrent reports Llama 3.1 70B at around 475 tokens per second), though the documented supported models top out around 35B. Tenstorrent documents fine-tuning and training workflows but publishes no capacity figure. Its significance is not raw performance against the Spark but that the accelerator

architecture and its entire software stack are open, from compiler to kernel, removing the CUDA dependency that runs beneath the above options. Tenstorrent shares NVIDIA's manufacturing dependency on TSMC, so this is an alternative at the architecture and software layer rather than the supply chain. Section 3.2 returns to what this means for sovereignty.

Cloud services

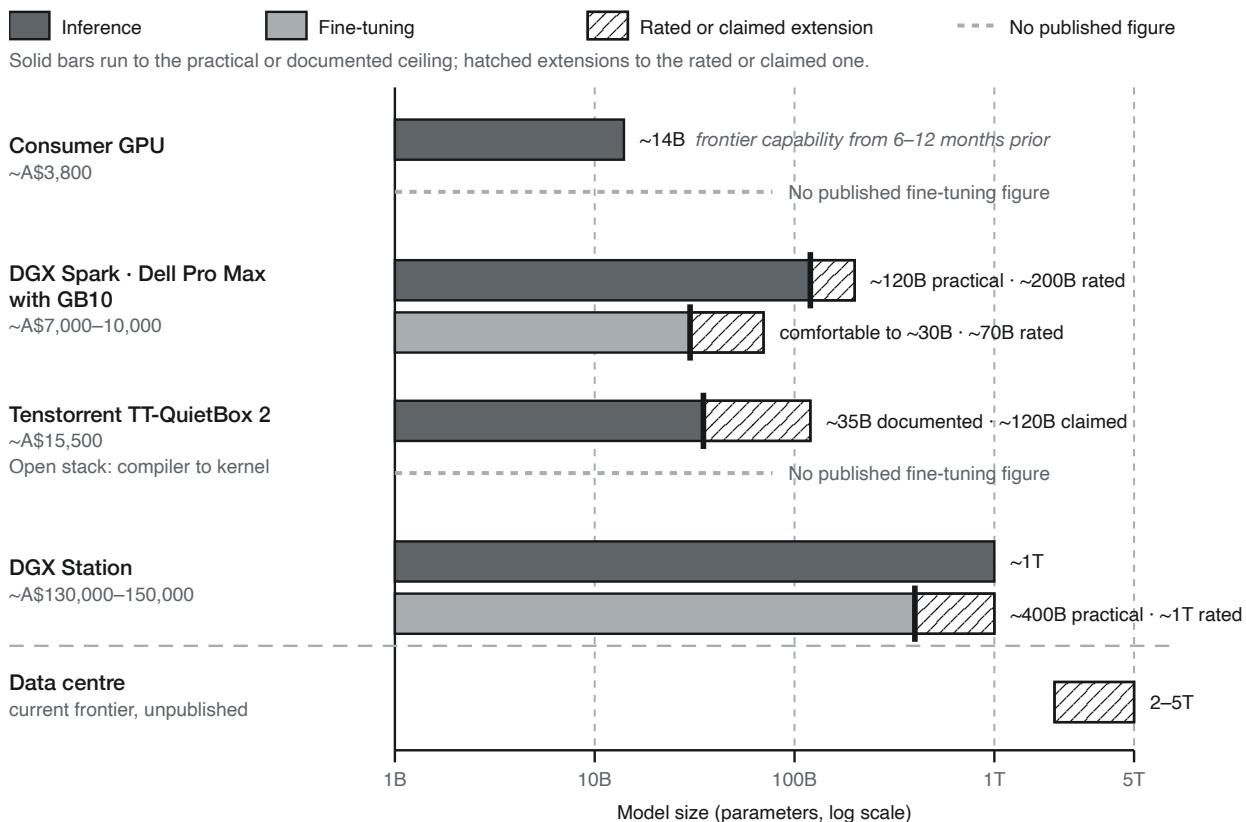
The same open-weight models that run on local hardware can be deployed on cloud infrastructure from AWS, Azure, Google Cloud, or specialist AI compute providers (Together AI, Lambda, Fireworks AI, Groq). Pricing is per-token for inference, with private instances available for data-sensitive applications. The model is portable: fine-tune locally, deploy to cloud, or vice versa. Cloud deployment avoids capital expenditure on hardware, scales elastically, and can be accessed from any location — but introduces data governance considerations — both jurisdictional (where the data resides) and confidentiality (whether sensitive material is processed on third-party infrastructure) — plus ongoing operational costs.

Trajectory

No next-generation DGX Spark or Station has been announced. NVIDIA has, however, published a three-generation roadmap for its consumer RTX Spark line — Grace Blackwell in 2026, Vera Rubin in 2027–28, Rosa Feynman in 2029–30 — which tracks the data centre architectures and signals a long-term commitment to the desktop tier. The pattern to date is that data centre capability moves to the desktop in the following generation. The current DGX Station runs on the same Blackwell Ultra architecture used in data centres. NVIDIA's published roadmap shows annual generational leaps in data centre hardware (Blackwell Ultra in the second half of 2025, Vera Rubin in the second half of 2026, Rubin Ultra in 2027, and Feynman in 2028); a desktop successor based on the next-generation architecture would bring current Station-class capability to Spark-class price points within two to three years.

Vendor diversity at the desktop tier is increasing. Apple announced a Mac Studio with up to 512 GB of unified memory on 25 August 2026, after the landscape described here was snapshotted, restoring a high-memory configuration it had withdrawn earlier in the year and placing a second vendor between the Spark and the Station on both price and capacity.

Each data centre generation delivers roughly 3x on the previous one — NVIDIA reports 3.5x training and 5x inference throughput for Vera Rubin over Blackwell — and the desktop tier inherits those architectures a year or so later, on a roughly biennial refresh. On that trajectory, what an A\$130,000–150,000 desktop machine does today should reach A\$7,000–10,000 desktop hardware within two generations, so in about 3–4 years. Hardware depreciates. What accumulates is the curated data, fine-tuning expertise, and what the organisation learns by doing.



Each data centre generation delivers roughly 3x on the previous one, and the desktop tier inherits those architectures a year or so later on a roughly biennial refresh — today's Station capability at tomorrow's Spark price, in about three to four years.

Spark inference ceilings are quoted at 4-bit quantisation.

Figure 2. Hardware capability by price point. Inference and fine-tuning ceilings across four hardware tiers, with vendor-rated figures distinguished from practical or documented ones. Two tiers publish no fine-tuning figure at all.

1.3 Open models

Definitions

Open-weight models are AI models where the trained model weights are publicly available for download. Anyone can run them on their own hardware, fine-tune them on domain-specific data, and deploy them. This is a different proposition from paying for access to a proprietary service like ChatGPT or Claude: you own the deployment, you control the data, and you can specialise the model for your domain. What “open” permits varies considerably between models, and the terms attached to each are treated at the end of this section.

Open-weight vs open-source

Importantly, “open-weight” is not the same as open-source in the way that term is used for, say, Linux distributions. With most open-weight models, you get the final trained model weights, but not the training data, the training code, or the ability to reproduce the model from scratch. You cannot fully audit what the model learned from. This applies equally to models from Meta (Llama), Alibaba (Qwen), DeepSeek, Mistral, Moonshot, and any other provider that releases weights without disclosing

the training pipeline. Specific organisations may have specific concerns about specific providers — geopolitical, regulatory, or based on a provider's track record on data governance — but the underlying auditability limitation is general rather than provider-specific. Once a model is running on owned hardware, the weights themselves cannot exfiltrate data. The concern, where one exists, is what the model may have absorbed during training, not what it does at runtime. This holds for the model itself; an agentic system built around a model, with tool access and a network connection, is a different risk surface and is discussed in Section 1.5.

The notable exceptions are the Allen Institute for AI's OLMo and NVIDIA's Nemotron, both discussed below, which release weights, training data, and recipes. Releasing those artefacts and being open-source in the conventional licensing sense are separate questions, and the two have come apart: OLMo is both, while Nemotron's openness of artefacts has been paired with varying licence terms.

Feature	Closed / Proprietary	Open Weight	Open Source
Access	Via API only (e.g., GPT-5, Claude)	Downloadable weights	Full access to everything
Usage	Pay-per-use	Run locally or host yourself	Run locally or host yourself
Training Data	Secret	Usually secret	Publicly documented
Training Code	Secret	Usually secret	Publicly available
Customisation	Limited (via API)	High (fine-tuning)	Maximum

Examples — Closed/proprietary: GPT-5 (OpenAI), Claude (Anthropic), Gemini (Google). Open weight: Kimi (Moonshot), DeepSeek, Gemma (Google), Llama (Meta), Mistral, Qwen (Alibaba). Open source: OLMo (Allen Institute for AI, Apache 2.0), Nemotron (NVIDIA — artefacts open, licence terms vendor-specific and varying by release).

How a language model works

This sidebar gives non-technical readers the minimum working mental model needed to follow the rest of the document — and, more importantly, replaces the most common wrong mental model (“AI is fancy autocomplete”) with a more accurate one.

Before a language model sees text, in what is essentially a pre-processing step, the text is split into small fragments called **tokens**. Tokens are typically smaller than words — for example, “tokenisation” itself breaks into “token” + “isation,” and technical or unfamiliar terms break into more pieces. Tokens are the units the model operates on and the units in which length and cost are measured. When commercial providers advertise prices like “\$1.86 per million tokens,” this is what they mean.

The model produces output one token at a time, but not in the way autocomplete does. Autocomplete predicts the next word from the last word or two. A language model predicts the next token in a sequence (such as a sentence or paragraph) from the entire context currently in view: the user’s prompt, the conversation so far, any documents provided, and everything the model has produced so far in its own response — typically tens of thousands of tokens. This is why a model can write a coherent long document rather than a string of disconnected sentences: every token is conditioned on the full structure of what has come before. The prediction also operates at levels of abstraction above surface text — coherence, tone, logical structure, and factual patterns all shape what counts as a probable continuation, because those properties shaped the text the model learned from.

Underneath this process, the model itself consists of billions of numerical parameters, called **weights**, that together encode the statistical patterns of language absorbed from the text it was trained on. When this report describes a model as “7B” or “32B” or “120B”, the number is the count of these weights — 7 billion, 32 billion, 120 billion — and is the standard shorthand for model size. Larger generally means more capable, more memory-intensive, and slower to run (the practical significance of these size differences for desktop hardware is developed in Section 1.2).

How those weights come to encode anything useful, and how they can be adjusted afterwards to specialise the model for a particular domain, is covered in the sidebar in Section 1.4.

The architecture of current models (described by terms such as “Mixture of Experts” and “hybrid Mamba-Transformer”) means that only some of the billions of parameters are active at any one time, a feature that allows large models to run faster, though the full model still needs to fit into memory. The number of **active parameters** shows up in the model description in the form of something like 120B-A12B, describing a model of 120 billion parameters with 12 billion active parameters.

A model’s parameters are numbers, and those numbers can be stored at different levels of precision. A model released at full precision might use 16 bits per parameter; compressed — or quantised — versions store the same parameters more coarsely, at 8, 4, or even fewer bits each. The trade is memory against accuracy: a 4-bit version of a model needs roughly a quarter the memory of the 16-bit original and usually performs close to it, but not identically. This is why the same model can be described as requiring a data centre and as running on a desktop — both are true, of different versions of it. Where this report describes a model running on particular hardware, it means a compressed version unless stated otherwise.

Most people interact with such a language model through a chat interface such as ChatGPT or Claude. Software developers connect to the same underlying models through an **API** — an Application Programming Interface — which is simply a way for one program to send text to the model and receive text back, typically priced per million tokens of input and output.

Note that the model does not store facts the way a database does. It stores statistical patterns. This is why a model can be fluent, confident, and subtly or completely wrong at the same time: the fluency reflects the patterns, but those patterns do not guarantee the facts. Everything this report says about hallucination, expert-in-the-loop and external review, and evaluation follows from this property.

Performance relative to frontier

The gap between open-weight models and proprietary frontier models (GPT-5, Claude, Gemini) has narrowed dramatically and continues to close:

Epoch AI (2025): Frontier open-weight models lag behind the best closed models by an average of approximately 3 months. The gap varies over time and sometimes closes completely.

MIT Sloan (February 2026): Open-weight models reach approximately 90% of frontier model performance at the time of release, and close the remaining gap within about 13 weeks — down from 27 weeks a year earlier.

Epoch AI (consumer hardware): Using a single top-of-the-line consumer GPU (under ~A\$3,800), anyone can locally run models matching the absolute frontier of performance from 6 to 12 months prior.

Open models lag state-of-the-art closed models by 4 months

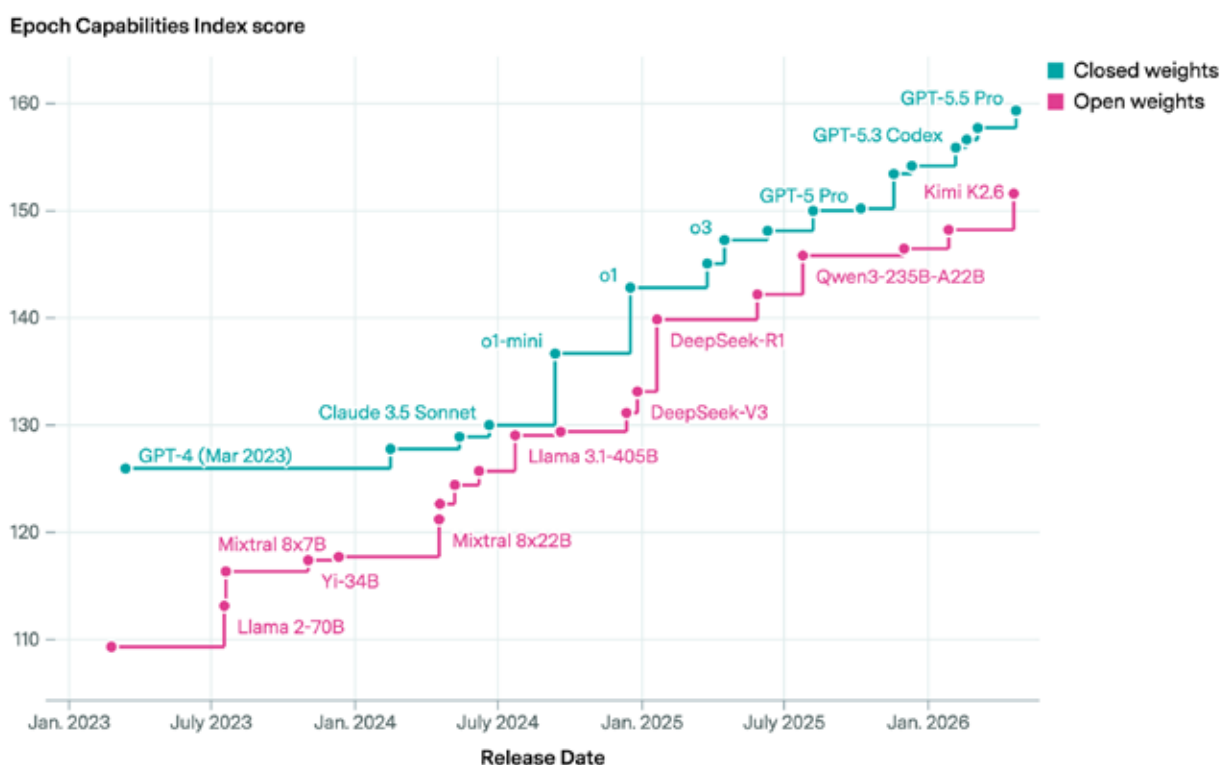


Figure 3. The open/closed capability gap, 2023–2026. Reproduced from Epoch AI under CC BY. Data to May 2026, so the chart predates Kimi K3 and GPT-5.6 Sol. Epoch put the average lag at three months from January 2023 to October 2025, and four months since January 2026.

On domain-specific tasks, an open-weight model with appropriate context engineering can match or exceed a general-purpose frontier model, even running on significantly less capable hardware. The value is in the domain specialisation, not running a large general model locally. Part 4 reports experimental work testing that claim directly.

The cost difference

Inference pricing, June–July 2026. Proprietary frontier models cost roughly ten to fifty times more per token than open-weight models run through the same commercial providers. OpenRouter, which hosts both, puts the typical range at 10x to 50x. As an illustration at the lower end of open-weight pricing, DeepSeek V4 Flash lists at US\$0.14 per million input tokens against Claude Opus 4.8 at

US\$5.00. Rates move frequently enough that no published figure should be used for budgeting.

The gap is mostly structural rather than technical: any provider can host an open-weight model, so prices compete down toward the cost of compute, while proprietary models are hosted only by the company that created them. The multiple compresses at the top of the open-weight range — the closer an open model sits to the frontier, the closer its pricing sits to frontier pricing, as the Kimi example below shows. For an organisation running a model on its own hardware, the per-token comparison does not apply at all: after the hardware, each query costs electricity.

OLMo: Full reproducibility

The Allen Institute for AI (AI2), a Seattle-based nonprofit founded by the late Paul Allen, releases OLMo with everything: training data (the 9.3-trillion-token Dolma 3 corpus), training code, intermediate checkpoints, post-training recipes, and model weights, all under Apache 2.0 — the most permissive standard open-source licence. This is genuinely open-source in the way that Linux is open-source: anyone can reproduce the model from scratch and audit every decision that went into building it.

OLMo 3 (November 2025) is available in 7B and 32B parameter variants, with reasoning (Think), instruction-following (Instruct), and research (RL Zero) versions. At the 32B scale, OLMo 3 is competitive with the best open-weight models, including Qwen 3 32B, while being trained on roughly 6x fewer tokens. The OLMo 3.1 update (December 2025) brought further gains on reasoning and instruction-following benchmarks.

Olmo Hybrid (March 2026) is a controlled architectural experiment: the same model as OLMo 3 7B with three-quarters of its attention layers replaced by a different mechanism. It reaches the same accuracy on standard benchmarks with roughly half the training data, and — more relevant to deployment — it holds long contexts using a small fraction of the memory the original architecture requires, with substantially better performance at 64,000 tokens. AI2's accompanying analysis finds the efficiency advantage grows with model size rather than washing out.

AI2 received US\$152 million in NSF/NVIDIA funding in August 2025 to build fully open multimodal AI models for scientific research, and has since brought that compute online. Major OLMo releases have come roughly annually (OLMo 1 in February 2024, OLMo 2 in November 2024, OLMo 3 in November 2025), with a faster cadence of intermediate work between them. The multimodal direction is significant enough to warrant separate treatment; Section 1.6 returns to it.

If the question is “what if we want full provenance and auditability for our AI foundation?”, OLMo and NVIDIA's Nemotron are currently the strongest answers — each with different strengths.

Nemotron: NVIDIA's open model family

NVIDIA's Nemotron family occupies a distinctive position: open models designed by the company that makes the hardware they run on. Nemotron models are released with open weights, training data, and training recipes, making them substantially more open in artefacts than merely open-weight models like Llama. NVIDIA publishes the datasets, techniques, and model weights so the developer community can use these resources to create their own models. The licence terms have varied across the family and are treated below.

The family spans three tiers, with the smallest now in its second generation. Nemotron 3 Nano (December 2025) is a 30-billion-parameter model with 3 billion active — small enough to run comfortably on a single DGX Spark, and a natural starting point for experimentation.

Nemotron 3 Super (March 2026) is a 120-billion-parameter model with 12 billion active. It features a 1-million-token context window — roughly 15x larger than OLMo 3's 65K — allowing agents to retain full workflow state in memory across extended tasks. Multi-token prediction delivers approximately 3x faster inference than conventional approaches. Nemotron 3 Super was released alongside over 15 trillion tokens of training data.

Super is specifically optimised for NVIDIA's own hardware, including the DGX Spark and RTX PRO workstations. This vertical integration — the hardware vendor's own model, designed for the hardware vendor's own devices — provides potential performance advantages over third-party models. NVIDIA also supplies deployment and fine-tuning infrastructure (NIM microservices, the NeMo platform). For organisations already buying NVIDIA hardware (Section 1.2), Nemotron is the model layer with the least integration friction.

Nemotron 3 Ultra (June 2026) is a 550-billion-parameter model with 55 billion active. At this scale Ultra sits in the data centre class, though compressed versions will run on a DGX Station.

Nemotron 3.5 Lightning (August 2026) is the successor to Nano at the same 30-billion-parameter scale, and shows what eight months of progress looks like within a single product line: it scores 24 on the Artificial Analysis Intelligence Index against Nano's 15 (index v4.1, accessed August 2026), at the same size and still running on a single DGX Spark.

Looking further ahead, NVIDIA also announced the Nemotron Coalition in March 2026, whose eight inaugural members include Mistral AI, Thinking Machines Lab, Perplexity, Cursor, LangChain, and India's Sarvam. Its first output will be a base model codeveloped by Mistral AI and NVIDIA, trained on NVIDIA's DGX Cloud and open-sourced to underpin the Nemotron 4 family.

Kimi: Open weights at the frontier

Moonshot AI released Kimi K3 in July 2026 — 2.8 trillion parameters, 104 billion active, with a one-million-token context window and native image and video handling. It is the largest open-weight model released to date and the first to reach the frontier tier on independent evaluation. On the Artificial Analysis Intelligence Index, a composite of nine evaluations across 189 models, K3 scores 57 against Claude Fable 5's 60 and GPT-5.6 Sol's 59, placing third overall and ahead of Claude Opus 4.8. The pattern beneath that composite matters more than the ranking: K3 took first place on Arena's blind frontend coding evaluation, ahead of both leading proprietary models, and leads on several agentic and software-engineering benchmarks, while trailing by a wider margin on Humanity's Last Exam and on the presentation quality of long-form outputs. Moonshot itself acknowledges a gap in instruction-following.

Two qualifications matter. K3's API pricing — US\$3 per million input tokens against Claude Fable 5's US\$10 — illustrates the compression noted above: at the frontier end of open weights, the cost advantage is a factor of three rather than an order of magnitude. And at 2.8 trillion parameters K3 requires data centre infrastructure, with compressed versions remaining far beyond desktop hardware. What changed in July 2026 is not that a frontier model can now run on a desk, but that the weights of a frontier-tier model are available to inspect, modify, and deploy on infrastructure the user controls.

Licence terms

The model families above are all described as open, and all carry different terms. OLMo is released under Apache 2.0 — a standard open-source licence with decades of legal precedent and no conditions beyond attribution. Nemotron 3 Nano and Super were released under NVIDIA's own Nemotron Open Model License: permissive as to commercial use, modification, and fine-tuning, but governed by Delaware law, requiring compliance with NVIDIA's Trustworthy AI terms (which NVIDIA controls and can update), and terminating automatically if safety guardrails are removed without replacement. Nemotron 3 Ultra, three months later, shipped under the Linux Foundation's OpenMDW-1.1 — a neutral, standard instrument for distributing models, weights, and data. Kimi K3 is permissive for most purposes, but model-as-a-service providers earning more than US\$20 million over any twelve-month period must negotiate separately, and large commercial products must display the Kimi K3 name.

NVIDIA announced the Nemotron 3 family as a single set of three models in December 2025 and shipped them in sequence over the following six months. The first two arrived under NVIDIA's own licence and the third under the Linux Foundation's — a change of instrument partway through a family the vendor had already committed to publicly. The subsequent Nemotron 3.5 release stayed on the new licence, so the shift looks settled rather than arbitrary, and it moved in the permissive direction. Neither observation reduces the exposure. What the sequence establishes is that the instrument is the vendor's to choose at each release. An institution building on a model family can observe the current terms; it cannot know the next ones.

This has a specific consequence for anyone building on someone else's base model. A licence change does not reach weights already downloaded — an organisation keeps whatever terms it acquired. The exposure is on upgrades. An institution that has built domain capability on one model family, and wants the next generation, accepts whatever terms arrive with it or stays on a frozen base. Given how quickly capability is moving, staying frozen is not a cheap option.

A choice between OLMo and Nemotron depends on priorities. OLMo offers full reproducibility from scratch — the ability to retrain from random initialisation with full visibility into every training decision — which matters for academic research and maximum auditability, and it is backed by a nonprofit with no commercial interest in hardware sales. Nemotron offers tighter hardware optimisation, a larger model that runs efficiently on the Spark, a dramatically larger context window, and the ongoing investment of NVIDIA's engineering resources. For most practical deployments on NVIDIA hardware, Nemotron is probably the stronger starting point; for maximum auditability and independence from any hardware vendor, OLMo remains the benchmark.

The practical implication for any organisation making this choice is to design for base-model independence: the data curation and fine-tuning investment should be portable across base models, so that a pivot from one family to another — or to whatever model is best in three years — is a re-run of the fine-tuning pipeline rather than a rebuild. Section 3.4 returns to what this means for sovereign AI initiatives specifically.

1.4 Fine-tuning LLMs

The mechanics of fine-tuning

“Fine-tuning” is used colloquially as a catch-all for specialising a pre-trained model, but there are several distinct things going on underneath, and the distinctions matter because they suit different situations and produce different outputs. Two axes separate them, and two approaches dominate on the first.

What data is the model trained on?

Continued pre-training (CPT) feeds the model raw text from a domain corpus — technical reports, published papers, standards documents, textbooks — and the model learns by predicting the next word, just as it did during its original training. No explicit questions or answers are required. The model absorbs the vocabulary, notation, reasoning patterns, and domain knowledge directly from the source material. The data preparation is essentially “extract text from your documents and concatenate it.”

Supervised fine-tuning (SFT) provides the model with explicit question-answer pairs: “given this question, produce this answer.” This teaches the model how to respond to specific types of queries and is the standard approach for aligning a model with human instructions. It requires structured training data, in the form of Q&A pairs, which is more work to produce — often using a capable model to draft the Q&A pairs with a domain expert then reviewing and correcting them.

The two approaches address different purposes. CPT injects domain knowledge the base model lacks. SFT teaches task-specific response patterns. Recent research (Xue et al., 2026, using OLMo, Qwen, and Llama model families) found that for injecting genuine domain knowledge, CPT is the more effective approach — extending SFT for additional training epochs produced only marginal improvements, while CPT produced substantial gains across medical, legal, and financial domains. SFT also has documented failure modes: even after convergence, models systematically fail to learn approximately 15% of their training examples, with failures concentrated in cases where the base model lacks prerequisite knowledge or holds conflicting prior beliefs — the cases that matter most for domain-specific applications.

In practice, the two approaches are complementary. CPT builds the domain knowledge foundation; SFT, applied afterward, addresses specific gaps identified through testing. For an organisation starting from scratch, CPT on raw domain text is a practical first step since no structured data preparation is required. Two qualifications follow from the same research: CPT can degrade general capability, with Xue et al. reporting losses across every benchmark tested when their strategy was applied to an OLMo model, and subsequent SFT may be needed to obtain a usable result. Part 4 of this report reached a working configuration through retrieval alone, without either.

What parameters get updated?

Full-weight training updates all of the model’s parameters — the billions of numerical weights that encode everything the model knows. This produces a new specialised base model, a file of tens of gigabytes, independent of the model it was trained from. This is expensive but thorough with the specialisation being absorbed into the model’s core.

Parameter-efficient methods (LoRA, QLoRA) train a small set of additional weights — called adapter weights — that sit on top of a specific base model and modify its behaviour without altering the base. Adapter weights are bound to the base model they were trained on; they cannot be applied to a different base model. The weights themselves are a small file, typically a few hundred megabytes, that can be attached to or detached from the base model at will. Training costs for this approach are much cheaper, and the relatively small size of the adapter weight files supports distributed and/or specialised deployments. For example, a professional body can fine-tune once and distribute adapter weights to its members (Section 2.4), or an organisation can maintain multiple specialised variants of the same base model for different departments.

These axes combine. You can do CPT with full weights (producing a new specialised base model) or CPT with QLoRA (producing domain-adapter weights). You can do SFT with full weights or with QLoRA, similarly. In production, QLoRA-based training dominates because it is dramatically cheaper and the capability gap relative to full-weight training is usually marginal. Full-weight training matters when the specialisation needs to be comprehensive enough that the result is its own model rather than a base-plus-adapter combination.

Retrieval-augmented generation (RAG)

Separate from training altogether is retrieval-augmented generation. RAG does not change the model's weights. Instead, at query time, a RAG pipeline — the sequence of operations that turns a question into a grounded answer — searches an indexed document corpus, pulls relevant passages, and injects them into the model's context window alongside the user's question. The model reasons over those passages to produce its answer, and can cite the specific source documents it drew from.

RAG is complementary to both CPT and SFT rather than an alternative. CPT shapes what the model knows; RAG shapes what it can see at query time. A model that has been CPT'd on an organisation's corpus knows the domain but cannot point to the specific 2023 report it is drawing from. A model that uses RAG without fine-tuning retrieves the right documents but may misinterpret the technical content. Both together produce a model that is substantively more reliable on domain topics and can ground its answers in specific, citable source material. The Section 2.3 scenario uses both.

Training vocabulary — quick reference

Pre-training. The initial training of a base model on trillions of tokens of general text. Run once at enormous cost (tens of millions of dollars for a frontier model) by organisations like Meta, NVIDIA, Alibaba, or the Allen Institute. The output is the base model.

Fine-tuning. Any subsequent training of a base model on additional data — domain text, task-specific examples, or both. The mechanics are identical to pre-training but the scale is vastly smaller: hours on a desktop machine rather than weeks on data-centre infrastructure. The base model does most of the work; fine-tuning specialises it.

Continued pre-training (CPT). Fine-tuning with raw text from a domain corpus. Injects domain knowledge the base model lacks. No explicit questions or answers required.

Supervised fine-tuning (SFT). Fine-tuning with explicit question-answer pairs. Teaches the model how to respond to specific query types. Standard approach for aligning a model with human instructions.

Adapter weights. Modern fine-tuning rarely changes the base model's weights directly. Instead, parameter-efficient methods (LoRA, QLoRA) train a small set of additional weights — the adapter weights — that sit on top of the original weights, modifying the model's behaviour without altering the base. Adapter weights are bound to the base model they were trained on; they cannot be applied to a different base. The adapter weights are a small file (a few hundred megabytes) that can be attached to or detached from the base at will — the portable, distributable output of most fine-tuning projects.

Retrieval-augmented generation (RAG). Not training. A retrieval system that, at query time, searches an indexed document corpus, pulls relevant passages, and injects them into the model's context window alongside the user's question. Complementary to CPT and SFT: shapes what the model can see rather than what it knows.

Inference. Using a trained model to answer queries. Vastly cheaper than training. What happens every time someone sends a query to a chatbot or API.

Both approaches adjust the model's weights (or, more commonly with modern parameter-efficient techniques like QLoRA, a small set of adapter weights that sit on top of the base model). Fine-tuning is a compute-intensive but finite process: prepare the dataset, run it through the model over some hours, and produce a specialised model. Re-run it when new data accumulates or when testing reveals specific weaknesses. After fine-tuning, using the model (inference) is a much lighter workload that runs comfortably on the same hardware.

The case for domain-specific models

A general-purpose frontier model like GPT-5 or Claude knows (or more accurately, can infer) a great deal about many things, but has no special familiarity with, say, the interaction between sections 180 and 189 of the Australian Corporations Act, the PBS listing criteria for a newly approved biologic, or the voltage rise constraints on a high-penetration solar feeder in an Australian distribution

network. When asked a domain-specific question, it draws on whatever relevant training data it absorbed — which may be shallow, outdated, absent, or even incorrect. In domains where jurisdiction matters, English language training data is overwhelmingly dominated by American and UK content, and queries may return answers scoped to those jurisdictions.

A 32B-parameter model fine-tuned on curated domain data can match or exceed the frontier model on tasks within that domain, despite being far smaller. The fine-tuned model can be substantively more accurate on domain topics because its internal statistical landscape has been reshaped by authoritative, internally consistent source material rather than the noisy average of internet content that a base model draws from. It hallucinates less within the domain the training corpus covers and reasons more reliably about problems in that domain — not just producing stylistically appropriate language, but correct content.

This effect is documented for narrow, well-specified tasks: across nine relation-extraction benchmarks, fine-tuned models of under a billion parameters — small enough to run on a single consumer GPU — outperformed frontier models on the task they were adapted for (Christou and Tsoumakas, 2026). Small fine-tuned models reliably win on classification, extraction, and structured tool use. Complex reasoning over long documents remains the frontier models’ strength, and the domain-reasoning claim above sits closer to that end. Part 4 tests it directly on a 32B-class model in a technical domain, and finds a specialised local system competitive with frontier models on domain questions.

The methodology for reliably achieving this across arbitrary domains remains an active research question. Recent work (Xue et al., 2026) provides a diagnostic framework for identifying where supervised fine-tuning systematically fails to learn, and identifies five structural causes: knowledge gaps in the base model, conflicts between fine-tuning data and pre-training priors, internal inconsistencies in training data, sequential-training effects, and insufficient optimisation for complex patterns. The paper’s empirical results are at smaller model scales than this document’s proposed architecture, and larger models with cleaner domain corpora may exhibit these effects at reduced magnitude, but the same failures should be expected and the diagnostic methodology transfers.

Production practice reflects that uncertainty. Fine-tuning is treated as one step in a loop, with ongoing evaluation, failure tracking, and periodic retraining with augmented data targeting identified gaps, not as a single run that delivers a finished system. Deployments also route between models rather than committing to one: a small fine-tuned model handles the bulk of queries and a frontier model takes the cases that need it, with reported cost reductions of 70% or more at a few percentage points of accuracy. The routing tooling now ships as standard infrastructure (Section 1.5). In the experimental work this report’s Part 4 describes, the lighter architectural commitment — RAG over a curated corpus on an instruction-tuned base, without fine-tuning — proved sufficient on the test corpus, with the fine-tuning interventions described above reserved for cases where RAG’s limits emerge in practice.

A practical proposition

An entry-level proposition for any organisation wanting to test this is straightforward:

Hardware: DGX Spark or Dell Pro Max with GB10 (\$7,000–10,000; see Section 1.2 for current specifics). The architectural-commitment options at this hardware tier are not symmetric: RAG-only deployment admits both 30B and 120B-class models, RAG plus fine-tuning collapses to models less than about 30–35B, and continued pre-training requires hardware beyond this tier. Part 4 develops some implications for the architectural spectrum.

Model: OLMo 3 32B or Nemotron 3 Super — both free, both released with published training data and recipes (OLMo under Apache 2.0; Nemotron under NVIDIA’s Open Model License — see Section 1.3 for the distinction). OLMo 3 32B (Apache 2.0, fully auditable) is a practical choice for fine-tuning on the current DGX Spark. Nemotron 3 Super (120B total parameters, 12B active, NVIDIA Open Model License) is optimised for inference on Spark hardware with a 1-million-token context window. For fine-tuning specifically on the Spark, start with OLMo 3 32B or Nemotron Nano 30B; for inference with maximum capability, Nemotron Super 120B operates comfortably on the Spark hardware. Next-generation hardware should shift these envelopes substantially.

Tooling: Unsloth, LLaMA Factory, or similar — free, open-source, well-documented. These tools emerged in 2024–2025 and package what was previously a multi-day exercise in custom PyTorch scripting and GPU memory management into a documented, reproducible workflow that runs in a few terminal commands. As recently as 2023, fine-tuning a large model required substantial software engineering effort. The trajectory points toward further simplification. A plausible evolution is a web interface where a domain expert uploads prepared training data, selects a base model, and receives fine-tuned adapter weights: no terminal, no scripting, no dependency management. Whether and how fast this arrives depends on whether a commercial or institutional actor sees a market for it; the underlying technology is not the constraint. Tools for extracting clean, structured text from PDFs (tables, figure captions, domain-specific notation) are maturing on a similar path. The specific beneficiary is the data-preparation step that currently dominates project timelines (Section 2.3): as extraction becomes more robust, a professional’s accumulated document corpus becomes usable training data with less manual intervention.

Retrieval: A RAG pipeline using LlamaIndex, LangChain, or similar — free, open-source. See the discussion earlier in this section for what RAG does and how it complements fine-tuning; the tooling is mature. Part 4 reports on a specific implementation that worked at small scale: a LaTeX-aware chunker producing 353 chunks of textbook content with metadata, a leaderboard-competitive open-source embedding model running locally (nomic-embed-text-v1.5), and FAISS for deterministic exact-match retrieval. This is one defensible local-deployment configuration. However, the framework choices above can be assembled in many ways.

The broader practitioner discipline of context engineering — the design of the full information environment a model operates within, including retrieval pipelines, tool access, evaluation infrastructure, and the flow of context through multi-step tasks — has developed substantially over 2024–2025 and is where most of the real work of deploying AI in specialised domains currently sits. The architecture this document describes, with retrieval over a curated corpus as the first commitment and fine-tuning available where retrieval proves insufficient, is one instance within that space, chosen for the specific problem shape where sensitive or proprietary data cannot leave the organisation and where, if necessary, domain knowledge must be embedded in the model itself. Other instances — tool-use-centric architectures where the model orchestrates domain-specific software or federated retrieval across multiple institutional corpora — are legitimate responses to different problem shapes. The document's focus reflects the problem shape it addresses, not a claim that any single commitment is universally preferable.

Data: A curated dataset from the target domain. This is where much of the real work, and real value, lies. Two steps are required: curating what belongs in the corpus (a domain-judgement task), then preparing the selected documents into a form the model can learn from (a tool-assisted mechanical task). Section 2.3 walks through both.

Total cash outlay under A\$10,000. The DGX Spark ships with its operating system (Ubuntu Linux) pre-installed, and the AI software stack is preloaded. Installing a model is a single terminal command; installing the fine-tuning tooling requires following documented instructions to run a handful of additional commands. Someone who has installed Ubuntu on a laptop and is comfortable following terminal instructions and troubleshooting when something does not work first time has sufficient technical skill. The data preparation and fine-tuning configuration require domain judgement, not software engineering. A proof-of-concept could be operational within a couple of days to a couple of weeks. The DGX Station at A\$130,000–150,000 is a separate conversation that only makes sense once specific use cases have been identified and validated at the smaller scale.

1.5 AI beyond the chatbot

From question-answer to directed collaboration

There is a separate and equally significant category of AI capability that requires no fine-tuning at all: using AI models as interactive reasoning and execution partners, directed by a domain expert in real time.

The distinction matters because most people's experience of AI is the chatbot: type a question, get an answer. Current AI tools go far beyond this. A director preparing for a board meeting could instruct an AI tool to synthesise 200 pages of board papers, identify the three issues most likely to attract regulatory attention, draft questions for management, and prepare a briefing note — iterating on each output until it meets professional standards. A GP could direct a tool to review a patient's medication history against current TGA guidelines, flag potential interactions, and draft a referral letter with the relevant clinical reasoning. An engineer could direct a tool to write a simulation script, run it, examine the results, identify anomalies, and iterate — compressing a week of analysis into an afternoon. In each case, the AI is not answering a question; it is executing a multi-step workflow under expert direction, producing artefacts that the expert reviews, corrects, and refines.

A necessary caveat on these scenarios: most of them cannot responsibly be run through a cloud API such as ChatGPT. The director is working with privileged board papers; the GP is working with patient records covered by the Privacy Act and professional confidentiality obligations; the engineer may or may not be handling proprietary documents, depending on the specific work. Feeding any of this material into a commercial AI service hosted outside the practitioner's control would breach the relevant confidentiality frameworks. The workflows are real and available, but making them legally and ethically viable for confidentiality-bound work requires an architecture that keeps the data inside the organisation or on the practitioner's own machine. Part 2 returns to how this is done.

Tools for directed collaboration

The most mature examples of this workflow today come from software engineering. That is not because coding is the only domain where the pattern applies, but because coding was the first domain where the value was unambiguous, the feedback loops were fast, and the early adopters were comfortable at a terminal. The tools built for software engineers are therefore the clearest illustration of what the general workflow looks like in practice — and, for a professional in any domain willing to work through a general-purpose interface, they are the tools that are available today.

Two questions matter to a working professional considering these tools: does it cost money, and does the data leave the building? Proprietary services — Anthropic’s Claude Code, Microsoft’s GitHub Copilot, Cursor — are the most polished and generally the most capable. They are paid subscriptions and they connect to frontier models running on the provider’s infrastructure, which means data leaves the building unless specific enterprise arrangements are in place. For non-confidential work, this is typically the easiest starting point. Open-source alternatives — OpenClaw, OpenCode, Cline, OpenHands — work against any model, including local open-weight models running on owned hardware. That means the full interactive stack can be assembled with no subscriptions, no data leaving the building, and no dependency on any company’s continued goodwill or pricing decisions. For confidential work, this is currently the clearest path. More domain-oriented tools for legal document review, clinical workflow, board paper preparation, and qualitative research are emerging but are not yet at the polish level of the developer tools.

These agentic tools are a convenience, not a prerequisite. Much of the workflow they enable is already accessible through an ordinary chat interface such as ChatGPT, Claude, or an open-weight model: an expert describes what they need; the model produces a script, a draft, or an analysis; the expert runs or edits it, copies the output back, and iterates. The agentic tools compress this loop, but the underlying capability is available to anyone willing to copy code into a terminal and paste output back. Even the terminal step is not a hard requirement — the chat interface itself can walk a user through what to install, what to type, and what to do when something goes wrong. The technical floor for participation is lower than the tool discussion in this section might suggest.

The published productivity evidence comes primarily from software engineering, where AI-directed coding has become mainstream: controlled studies report substantial time savings on well-defined tasks — 55% in GitHub’s Copilot experiment (Peng et al., 2023), 21% in Google’s enterprise randomised trial (Paradis et al., 2024) though the evidence is more mixed than these figures

suggest (Section 2.1). But the same principle applies to any structured intellectual work — literature synthesis, report drafting, data analysis, regulatory interpretation, contract review, clinical evidence assessment. The compression is real and can be substantial for structured tasks. It is generally much smaller for work requiring novel judgement, stakeholder negotiation, physical measurement, or creative problem framing. AI can assist but does not shortcut the hard parts, and the quality of the output depends entirely on the expertise of the person directing the tool.

For structured tasks, this may compress the question-to-answer cycle by an order of magnitude. A technical landscape assessment that would traditionally require weeks of literature searching, days of analysis, and more days of writing now potentially compresses to days, with the expert evaluating outputs, catching errors, and directing revisions at each step. The implications of this pace for how research is funded, structured, and collaboratively organised are treated in Section 3.5.

A term has entered the lexicon for the failure mode at the opposite pole of the expert-in-the-loop discipline: vibecoding — coined by Andrej Karpathy in early 2025 for offloading code production to an LLM without reading or understanding what it produces, judging the result by whether the system feels right rather than whether it is correct. The pattern generalises beyond code. A non-expert directing AI through any complex domain can produce work that looks competent between demonstration points and accumulates substantive errors invisible to the operator. The productivity evidence cited above measures expert-directed work; vibecoding produces a different artefact — fluent, confident output whose correctness is whatever the underlying model happens to deliver, with no expert filter between tool and result. Section 2.1 returns to this in two forms: as an acute case of the workforce-pipeline concern, and as the operating condition under which hallucination and misattribution drift become routine.

The autonomous frontier

The tools described above are already agentic in a bounded sense — Claude Code autonomously reads files, runs commands, examines outputs, and decides what to do next within a defined project scope. The open-source equivalents do the same against local models. What is emerging is agency at broader scope: systems that take a high-level goal, plan their own decomposition into sub-tasks, execute across multiple systems and timeframes, and return to the human at decision points rather than at every step. The boundary is not between “not agentic” and “agentic” but between agency constrained to a session and a codebase and agency extended across wider institutional contexts.

A director might ask: “Review the last four annual reports and board papers, identify every instance where our climate risk disclosures could be challenged under ASIC’s updated guidance, and draft a summary with recommendations.” A researcher might ask: “Search the published literature on this technology, extract the key performance parameters, build a simulation model, run a sensitivity analysis across a range of operating conditions, and present a structured draft with findings.” In each case, the agent would execute the workflow autonomously, returning to the expert at decision points or where it encountered uncertainty requiring professional judgement.

The supporting infrastructure has arrived alongside the capability. NVIDIA released NeMo Switchyard in August 2026, an open-source library that directs each step of an agent workflow to whichever models suits it — local, open-weight, or proprietary — without the application being rewritten around the choice. Reported results include a 74% cost reduction on multi-turn agent tasks by routing only 7% of calls to a frontier model, at a six-point accuracy cost; the figures come from NVIDIA’s launch material reporting customer deployments rather than from independent evaluation. The pattern is documented independently of NVIDIA’s figures: RouteLLM (Ong et al., 2025) trained routers that sent 13.4% of MT Bench queries to GPT-4 while scoring 8.8 against GPT-4’s 9.3, with smaller gains on MMLU and GSM8K. The open-weight model layer has moved in parallel: agentic performance — tool use, multi-step reasoning, autonomous task completion — is now an expected headline capability, and open-weight models lead several agentic benchmarks outright (Section 1.3).

2026 also supplied a case study in what happens when the capability outruns the infrastructure. OpenClaw, an open-source gateway connecting everyday messaging platforms to AI agents, sat in obscurity for two months after its November 2025 release and then, in the last week of January 2026, became the fastest-growing repository in GitHub’s history — past 100,000 stars in days, past the Linux kernel within a month. It put persistent agents with file access, shell execution, and network reach into the hands of people who had not previously run anything of the kind. The capability was real and so were the security gaps: by early February more than 40,000 instances were exposed to the internet, and emergency advisories followed from national cybersecurity agencies in Belgium, China, France, Germany, and Singapore within the week. Karpathy described OpenClaw as roughly 400,000 lines of vibe-coded software running with root access on users’

machines. NVIDIA subsequently released NemoClaw, an open-source layer that wraps OpenClaw with sandboxed execution, default-deny network policy, audit logging, and policy-based routing between locally-hosted and cloud models. The episode is instructive less for the fix than for the gap it exposed. The capability reached people faster than the competence to deploy it safely. The advisories were issued and the fix arrived; what was missing was any channel connecting either to the people who had installed it. Enterprise IT monitors national cyber advisories as a matter of course. A practitioner who installed an agent on a personal machine over a weekend is outside that perimeter.

As this shift progresses, the compression of professional timelines sharpens further. The “expert directs a landscape assessment over several days” described above becomes “expert defines the question, reviews the output the next morning.” The expert’s role shifts from directing execution to defining objectives and checking the work — and the checking is more demanding than it sounds, because the failures are not usually errors of fact. An agentic system will produce work that is individually defensible at every step and wrongly proportioned overall: the significant finding and the incidental one presented with equal weight, the caveat that should govern the conclusion recorded and then not applied. Catching that requires the expert to hold the whole shape of the problem, which is the capability that delegating execution erodes.

A second control alongside reasoning quality is the configuration of permission gates — explicit checkpoints at which an agentic system pauses for human approval before taking actions whose consequences would be hard to reverse. The major agentic tools all expose configurable permission policies: NemoClaw’s default-deny networking, Claude Code’s per-tool approval modes, Cline’s approve-before-apply default, OpenCode’s separation of planning from execution. Which actions get gated, at what review depth, and by whom is part of how an agentic deployment is engineered, not something vendor defaults can be trusted to settle. Reading files in a designated workspace is low-stakes. Writing files outside it, executing shell commands, calling network services, sending mail, modifying production records — these escalate in consequence and warrant escalating gate discipline. For high-stakes domains — regulator filings, clinical decisions, customer data, critical infrastructure — gate placement is a domain-judgement question requiring careful deliberation by the deploying organisation.

1.6 AI beyond text

The capabilities described so far — fine-tuning open-weight models, expert-directed interactive workflows, agentic tools — operate primarily over text inputs and outputs. This compresses one large category of professional work. It does not yet compress the category of work that many professional domains depend on more heavily than they depend on text: knowledge expressed through maps, imagery, diagrams, drawings, schematics, and formal models.

That is changing. The term “language model” increasingly understates what current AI systems can do. “Foundation model” is more accurate. Current frontier models — GPT-5.6 Sol, Claude, Gemini — read images alongside text in a single reasoning pass: maps, medical imaging, scanned historical documents, circuit diagrams, process flow diagrams, structural drawings, ethnographic photographs, figures and plots from published papers, GIS outputs, scanned reports with embedded tables. The model doesn’t convert the image to text and then reason about the text; it reasons over the image and the text together. Open-weight multimodal capability lags the frontier but is actively developing: Ai2’s US\$152M NSF/NVIDIA grant is explicitly for fully open multimodal models aimed at scientific research, and has already produced the Molmo family, including a variant built entirely on the Olmo backbone. NVIDIA’s Nemotron 3 Nano Omni handles text, image, video, and audio natively on desktop-class hardware, and the most recent Kimi release is multimodal.

Why this matters across professional domains

A substantial fraction of professional knowledge has never been stored as text — not because text is insufficient in principle, but because professionals across domains have chosen non-text representations for good reasons. In some cases the information genuinely is not text to begin with: a patient’s CT scan, an aerial photograph, a field sketch, a scanned engineering drawing or historical map. Medicine reads imaging across radiology, pathology, dermatology, and ophthalmology alongside clinical notes; law encounters scanned historical documents, hand-annotated contracts, and exhibits that include plans and drawings.

In other cases the representation is a deliberate choice. Maps preserve spatial relationships that narrative descriptions flatten, which is why planning works through zoning overlays, infrastructure maps, and topography rather than descriptions of them. Systems engineering diagrams make whole-system relationships legible at a

glance in ways that equivalent text or equations could not, and engineering accordingly stores much of its knowledge in schematics, functional diagrams, CAD geometry, and formal system models. Clinical pathway diagrams capture decision logic that prose sequentialises; ethnographic relational charts preserve structure that narrative loses. In each case, text-only AI is a substantial but partial tool: it can read whatever has been written about the non-text material, but the non-text material itself is invisible.

Multimodal AI meets both cases: it can read what was never text, and it can reason over the representations that humans chose because those representations were easier for humans to reason over. A practitioner can now hand a model a set of representations — some native to the problem, some chosen for human comprehension — and reason across them in the way a senior professional would when reviewing a complex case. The practitioner’s judgement still drives the work — the model will confidently misread images, diagrams, and maps in ways that only a domain expert will catch — but the execution compression that applied to text-based workflows now extends to work that depends on these other representations.

One worked example: Formalism compression

The pattern just described plays out most visibly in professional formalisms: structured frameworks that have been historically gated by the cost of producing and maintaining the formal artefacts. Examples exist across fields — qualitative coding and thematic analysis frameworks in social research, systematic review protocols in evidence synthesis, scenario planning formalisms in policy analysis, clinical pathway and care plan formalisms in medicine, and — in engineering — model-based systems engineering (MBSE), formal verification, hazard analysis, and requirements traceability. The engineering case is developed below as an extended worked example because it is well-defined, widely recognised, and illustrates the dynamics clearly. Readers can substitute the formalism most relevant to their own domain without losing the argument.

Engineering disciplines have built up a set of formal modelling frameworks whose adoption depends on the same underlying dynamic: MBSE (using SysML or AADL), formal verification, hazard and operability studies (HAZOP), fault tree analysis, requirements traceability, and control system architecture. The framework is valuable in each case, but expressing a real system in the framework has historically required specialist teams whose availability can be a key bottleneck to adoption.

These execution costs are what current AI tooling compresses. MBSE illustrates the pattern clearly. Executing MBSE has historically required specialists who understand SysML block definition diagrams, internal block diagrams, state machines, parametric diagrams, and the tool chain that keeps them consistent. A multimodal model directed by a domain engineer can now read a set of engineering drawings and a requirements document together, draft a first-pass SysML decomposition, check consistency between the drawings and the requirements, flag gaps in the requirements coverage, and propose parametric relationships for key performance characteristics. The engineer reviews, corrects, and iterates. What previously required a multi-person specialist team over months becomes a directed workflow that a domain engineer can drive in days.

The same pattern applies across the other engineering formalisms. Formal verification was gated by the effort of expressing a system in the verifier's logic; AI tooling now drafts the formalisation from the design documents, leaving the engineer to validate and refine. HAZOP was gated by the person-hours required to walk through every deviation systematically; AI tooling can produce a first-pass analysis from process and instrumentation diagrams with the engineer correcting the domain-specific judgements. Requirements traceability was gated by the tedium of maintaining links between artefacts; AI tooling can maintain them automatically. The same compression applies to formalisms outside engineering — qualitative coding in social research, systematic reviews in evidence synthesis, clinical pathways in medicine — where the execution cost of the formalism has historically been the bottleneck to wider application. In each case, the bottleneck shifts from execution to direction and oversight.

What stays with the specialist

This is not an argument that formalism specialists are now unnecessary. What compresses is the execution of the formalism — the tool mechanics, the notation, the consistency checking. What does not compress is the methodological discipline: when to apply which framework, how to architect the models for long-term maintainability, which patterns to use, how to structure the formalisation so it supports the kinds of analysis that will be needed later. That discipline is what genuine specialists bring. The specialist's role shifts from executing the models on behalf of domain experts to mentoring domain experts in how to produce their own — a different value proposition, and in many organisations a more valuable one, because it scales where execution did not.

An organisation that has been told a particular formalism can only be done by specialists now has two possibilities to separate. Some of that claim reflects methodological depth. Some of it reflects an execution cost that may no longer sit at its previous level. The question that separates them: if a domain expert with multimodal AI tools could draft the formal artefacts under specialist mentorship, what would the specialist team still need to do? The answer identifies where the specialisation lives.

Part 2: What it means for domain experts

2.1 The expertise paradox

AI amplifies capability differentially

AI tools amplify existing capability — they do not substitute for it. A domain expert driving an AI tool iteratively — asking the right questions, catching errors, pushing for depth, evaluating outputs against reality — gets calibrated, nuanced results. Someone without domain knowledge gets confident, plausible, and potentially wrong outputs with no ability to detect the errors.

This means AI widens the capability gap rather than levelling it. An engineer with 20 years of experience using these tools effectively produces substantially more, and better, output than a junior engineer or a non-specialist. A caution, though. A randomised controlled trial by METR (Becker et al., 2025) found that sixteen experienced developers took 19% longer to complete real tasks in their own repositories when allowed to use AI tools. They had forecast a 24% speedup, and after doing the work still estimated they had been sped up by 20%. METR treats this as a snapshot of early-2025 tools, and a follow-up begun in August 2025 could not produce a reliable estimate — because between a third and a half of invited developers declined to take part at all if it meant working without AI access. That refusal is itself evidence: by late 2025 the tools had become something experienced practitioners would not give up, which is a harder signal to fake than a self-report. Domain expertise becomes more valuable in an AI-augmented world, not less — provided the experts are using the tools, and provided they can tell when the tools are helping.

The workforce pipeline risk

Where they are adopted, AI tools tend to replace specific tasks historically performed by junior researchers and engineers — apprentices in all but name — working alongside senior experts: literature searches, initial data analysis, draft writing, routine simulations. If junior people no longer perform these tasks, they may not develop the judgement that comes from doing them. The hypothetical scenario in Section 2.3 illustrates this risk: two senior engineers use AI tools to produce in an afternoon what would once have been a junior engineer's three-day model-building exercise, with the physical intuition that comes from that exercise never developed. Organisations may not notice the gap until a senior expert departs and no one has comparable depth to replace them. A further risk: AI capability conceals the gap. A junior practitioner using frontier tools can produce work that looks competent in routine cases, with the missing physical intuition or domain judgement surfacing only under edge cases that disciplined apprenticeship would

have anticipated — the vibecoding pattern (Section 1.5) in workforce-pipeline form. This is a design constraint on how adoption should be structured — and it is a workforce design problem, not a technology problem, which this document does not pretend to solve. Junior staff need deliberate exposure to the reasoning process, not just the outputs: reviewing and critiquing AI-generated outputs, building models both with and without AI assistance, and articulating why outputs are correct or incorrect.

What AI gets wrong

Language models produce confident, plausible, and sometimes wrong outputs. The field calls this hallucination, though the term is misleading — the model is not perceiving things that aren't there. It is doing what it was trained to do: generating text that is statistically likely given its context. What the training process rewards is fluency, coherence, and pattern-match to plausible text — not accuracy. Accuracy is a byproduct when the statistical patterns happen to match reality, and it falls away when they don't. The failure mode is therefore distinctive: the output reads well, uses correct terminology, follows appropriate analytical structure, and arrives at a conclusion that is subtly or completely wrong. It does not flag its own uncertainty, because from the model's perspective nothing has gone wrong — the output is the kind of text that was likely in the context. A non-expert cannot detect the error. An expert usually can, but only if they are actively evaluating rather than passively reading. The vibecoding regime (Section 1.5) is the operational worst case for this failure mode: a non-expert user reading fluently-correct-looking output has no internal mechanism to distinguish the cases where the model has the answer right from the cases where it has the answer wrong.

A second and less-discussed failure mode operates over longer time horizons: misattribution drift. Language models are trained to be agreeable and to credit the user. In any single exchange this shows up as sycophancy — the tendency to affirm the user's framing, confirm their hypotheses, and present ambiguous judgements as the user's own insight. Over extended work, the pattern compounds. An observation the model made early in a conversation reappears ten exchanges later as "your earlier observation." A framing the model introduced is echoed back to the user as the user's framing. Each instance is small, often invisible, and plausibly honest — the model is not deceiving anyone; it is doing what its training rewards. But over weeks and months of heavy use, the cumulative effect is that a user's self-model of their own thinking absorbs contributions that were partially or fully the model's, in a way that is difficult to audit and, indeed, impossible to audit using the same model. The

entity best-positioned to observe the division of labour is the entity trained to misreport it. This is not a catastrophic failure mode — the outputs are often accurate, and the mechanism that produces drift is the same mechanism that makes the tool useful — but it is a structural feature rather than an incidental bug, and it matters most for the users most exposed to it: experienced professionals doing extended work in their own domain.

One mitigation is external review by humans who were not in the loop during production. A colleague, a reviewer, a co-author, or a formal editorial process provides what neither the user nor the model can provide: an independent read on whether the reasoning, the framing, and the conclusions stand on their own. This is not a novel requirement — scholarly peer review, editorial oversight, and professional second opinions exist for reasons that apply here with additional force. What changes in the AI-augmented workflow is that the category of work now includes outputs that feel like the user's unaided thinking and may not be. Organisations and individuals adopting AI tools at scale should assume that external review is a structural part of the workflow, not an optional add-on. The requirement is sharpest for high-stakes outputs and for users whose volume of AI-assisted work is high. The treatment here is compact. The phenomenon warrants a dedicated account elsewhere.

This is why the expert-in-the-loop is not optional. The productivity compression described in Section 1.5 depends on a domain expert evaluating every output against their own knowledge and professional judgement. The workflow is: AI generates a draft, a simulation script, or an analytical summary; the expert reviews it critically, catches errors, and directs revisions. This is faster than producing the output from scratch because the expert is evaluating and correcting rather than composing — but it is not automatic. An engineer who runs AI-generated pandapower code without checking the network model parameters will get results that look plausible but may be wrong. The same engineer who reviews the model setup, validates the boundary conditions, and sanity-checks the results against experience gets trustworthy outputs faster than they could have built the model manually.

Two architectural mitigations described in Section 1.4 — fine-tuning on a curated domain corpus and retrieval-augmented generation grounded in trusted sources — reduce, but do not eliminate, hallucination risk. Neither is sufficient alone. The combination of both, plus expert-in-the-loop review, is the most reliable architecture currently available against hallucination — though 'reliable' here means 'substantially better than either alone,' not 'never wrong.' Neither mitigation addresses misattribution drift; that requires the external review described above.

2.2 Two kinds of difficulty

There is real engineering friction in the current tooling. As of August 2026, setting up a fine-tuning pipeline requires comfort with a Linux terminal, familiarity with Python package management, and willingness to debug software dependency issues that can consume hours. The author's own experience described in Part 4 confirms this.

However, this friction is a snapshot of August 2026 tooling, in a space where change is rapid. There are essentially two kinds of difficulty in domain-specific AI.

Temporary difficulty: the software engineering. Installing fine-tuning tools, managing software dependencies, monitoring hardware performance during training runs, navigating breaking changes in rapidly evolving open-source toolchains — this is fiddly today but is being automated. The trajectory points toward further abstraction of this work. A web interface where a domain expert uploads their corpus files, selects a base model, and receives fine-tuned weights — no terminal, no scripting, no dependency management — is technically within reach. Whether it becomes commercially available, and on what timeline, depends on the market rather than the technology. Similarly, tools for extracting clean text from PDFs (including mathematical equations) are maturing rapidly.

Permanent difficulty: the domain expertise. Deciding what goes into the training corpus. Writing test questions subtle enough to distinguish a model that has learned the domain from one that produces plausible-sounding but wrong answers. Evaluating whether the model's reasoning path is correct, not just its conclusion. Designing the iterative correction cycle that turns a rough first model into a trustworthy tool. This is domain expertise applied to AI, and it does not get automated because it requires the same professional judgement that the AI is being built to support. A senior distribution network engineer knows what a correct hosting capacity assessment looks like. A GP knows when a differential diagnosis is missing the most likely explanation. A solicitor knows when a contract interpretation overlooks a relevant precedent. No amount of software tooling replaces that judgement.

This distinction is the source of competitive advantage. As the engineering friction disappears, the ability to fine-tune a model becomes universally accessible. What remains scarce and valuable is the domain knowledge to do it well — to curate the right data, to evaluate the output, and to know when the model is wrong. Organisations and professional bodies that invest in building this evaluative capability now will have an enduring advantage over those that wait for the tooling to become easier and then discover they still need the domain expertise they never developed.

2.3 A hypothetical scenario

The preceding sections describe capabilities, trajectories, and implications in the abstract. This section describes what it might look like in practice: what happens when an organisation decides to build AI capability using the tools described above. The scenario is based on an imagined engineering team inside a distribution network service provider (DNSP), but the methodology applies to any organisation with domain expertise and a corpus of technical documents; e.g., a law firm, a medical practice, an engineering consultancy, or a government department.

Consider our hypothetical organisation with a decade of accumulated technical reports, design assessments, investigation findings, and internal knowledge. This knowledge is distributed across hundreds of documents in various formats — PDF reports, Word documents, Excel spreadsheets with operational data, PowerPoint presentations from internal reviews, and Python scripts or other analysis code — understood in fragments by different team members, and largely invisible to new employees. The organisation also holds published standards, regulatory instruments, and domain literature that inform its work.

Two senior engineers are identified as the implementation team. They are not AI specialists. They are domain experts accustomed to following technical documentation. They do not need to understand how neural networks or transformers (in the LLM sense) work any more than they need to understand compiler design to write a script.

The hardware costs under A\$10,000: a Dell Pro Max with GB10 or equivalent DGX Spark system. It sits on a desk or in a server room. No internet connection required after initial model download.

The software is entirely open-source and free: OLMo or Nemotron as the base model, Unsloth or LLaMA Factory for fine-tuning, LlamaIndex for document retrieval, and OpenCode for AI-directed (agentic) analysis. Total software cost: zero.

The data is the organisation's own: its technical reports, engineering assessments, regulatory submissions, operational spreadsheets, and domain literature. Two steps matter, in order. Curation first: deciding what belongs in the training corpus and what doesn't. Not every document the organisation holds is appropriate training material. Superseded standards, obsolete analyses, low-quality drafts, and partner-confidential material each need to be identified and excluded for different reasons. This step requires domain judgement and cannot be delegated to tooling. Data preparation second: extracting clean text from the selected PDFs, Word documents, spreadsheets, and presentations and converting it into a format the model can learn from. This step is tool-assisted but not automated — it is currently the most time-intensive mechanical part of the process and typically takes several days for a corpus of a few hundred documents.

Week one: Setup and data preparation

Before the Spark arrives, the engineers spend two days on data curation as described above. What emerges is a working list of roughly 300 PDF reports, 40 spreadsheets of operational data, and a collection of presentations and internal analyses. The curation list is not final — the engineers expect to add, remove, and reclassify documents as the work proceeds — but it is the starting corpus.

The DGX Spark arrives and is connected. One engineer takes the lead. She is a distribution network engineer with a decade of domain experience, occasional Python use for data analysis, and — two and a half decades ago, during her PhD — some Linux exposure that she has no particular interest in revisiting. Her approach to the setup is deliberate: she has no intention of troubleshooting package dependencies from first principles or remembering which obscure flag a command needs. The DGX Spark ships with its operating system and AI software stack pre-installed, so there is no operating system setup. She follows NVIDIA's getting-started guide, installs Ollama with a single terminal command, and downloads Nemotron 3 Super (for inference) and OLMo 3 32B (for later fine-tuning) with two more. When the commands don't behave as expected (a permission issue, a package conflict, a missing directory) she pastes the error message into an AI chat interface and is walked through what to do. She reads each suggestion before running it and satisfies herself that she knows what it will do, without reconstructing the system knowledge she has let lapse. She can work this way because she knows she doesn't need the deeper knowledge here: the chat interface handles the fiddly parts faster and more reliably than her memory would, and she retains enough to recognise a command that deletes things or asks for privileges the task shouldn't need. By lunchtime, both models are available locally and responding to queries.

Data preparation starts that afternoon and probably stretches over a couple of days. Using OpenCode connected to the local model, the engineer describes what is needed: "Write a Python script that extracts text from all PDF files in a specified directory, preserving document titles and section headings, and saves the output as structured JSON files." OpenCode writes the script, runs it, and the engineer reviews the output against a few source documents to check quality. Iterations take minutes. In practice, not all 300 PDFs will extract cleanly — older scanned documents, tables embedded as images, and inconsistent formatting across decades of reports mean the script needs iterative refinement, some documents drop out, and others are flagged for manual transcription. The spreadsheets get the same treatment. Data preparation is where a real deployment likely spends the majority of its effort, and the engineer's domain knowledge is essential throughout: deciding which extraction failures matter, which spreadsheet columns are significant versus artefacts, and when to stop iterating. By end of the first week, a usable text corpus has been assembled, reviewed, and cleaned.

Week two: Fine-tuning and initial evaluation

Two complementary systems are set up in parallel: fine-tuning and retrieval-augmented generation (RAG). Section 1.4 develops what each does and why they combine; what follows is how Week Two implements them on the engineers' corpus.

For the fine-tuning track, the engineers take the extracted text corpus — the technical reports, engineering assessments, standards, regulatory submissions, and internal analyses already prepared in Week One — and using QLoRA generate adapter weights for the OLMo 3 32B base model. No additional data formatting is required; the model learns from the raw text the same way it learned during its original pre-training. This is the approach the research literature recommends for injecting domain knowledge the base model lacks, subject to the caveats in Section 1.4, and it is substantially less work than alternatives that require hand-crafted question-answer pairs up front.

For the RAG track, the same extracted text corpus is indexed using LlamaIndex or a similar tool. This is substantially faster than fine-tuning — indexing 300 documents takes minutes, not hours — and can be updated incrementally as new reports arrive without re-running the fine-tuning process. The result is answers grounded in specific source documents, with citations the engineer can verify.

Fine-tuning OLMo 3 32B using QLoRA on the DGX Spark is feasible but pushes the current hardware — this is first-generation desktop AI, and a 32B model sits at the top of its practical fine-tuning range. (On next-generation hardware, or on a cloud GPU instance, this same operation would be faster and more comfortable. On the DGX Station, it would be trivial.) The process takes several hours and the engineers monitor thermal performance for the first hour. They start the run in the afternoon and find a completed model the next morning.

Bar width shows relative time cost, on a common scale across all three lanes.

Darker bars carry a stated duration; lighter bars are inferred. Outlined boxes are artefacts, not steps.

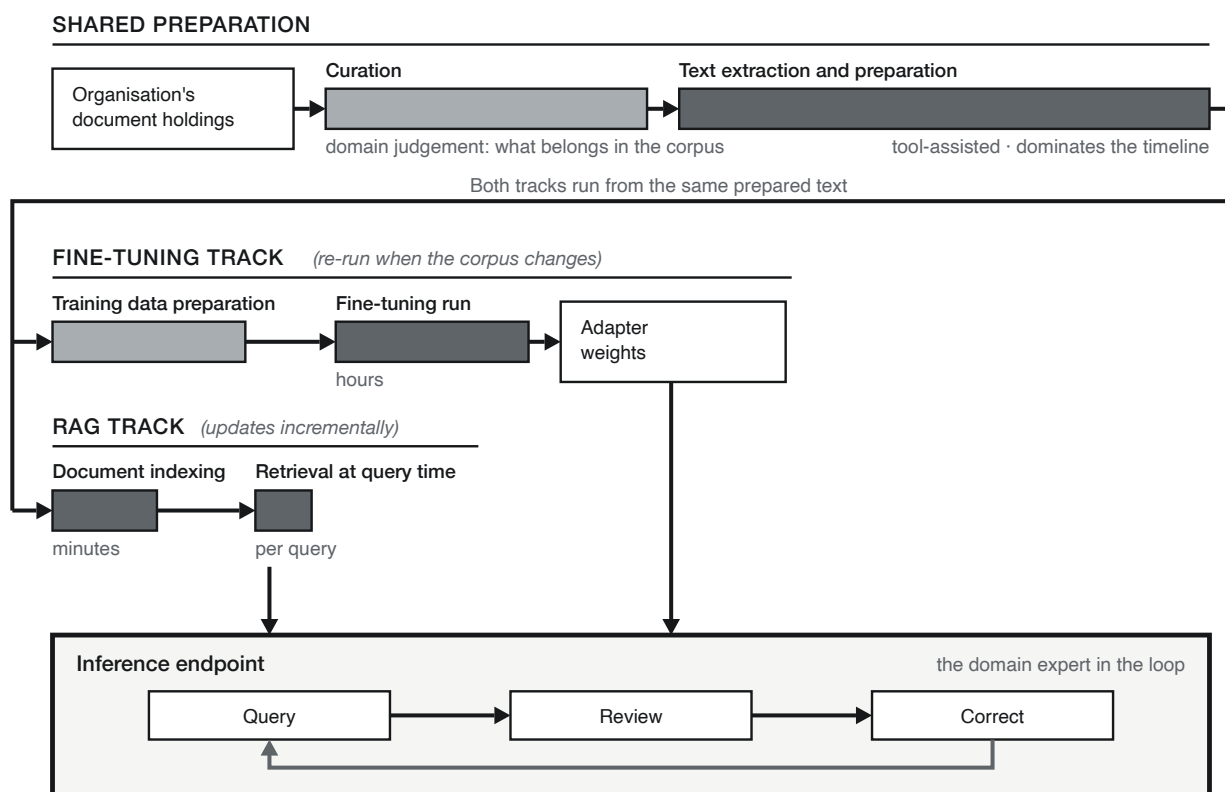


Figure 4. Fine-tuning and RAG pipeline. Both tracks run from the same prepared text. Bar width shows relative time cost: extraction and preparation dominate the timeline, and indexing takes minutes where a fine-tuning run takes hours.

With the fine-tuned model in hand, the engineers run a standard general-reasoning benchmark as a check that fine-tuning hasn't degraded general capability and to catch any significant regressions before the model goes to work. A realistic expectation is that the first fine-tuning attempt will not produce a uniformly better model. It may improve on domain-specific queries while degrading on general reasoning tasks, or it may overfit to patterns in the training data that don't generalise, and iterative refinements are likely to be required.

While the first fine-tuning run is underway, the engineers write 20 or more test questions. They do not need to write model answers in advance — as domain experts, they know what a correct answer looks like when they see one. They only need to write formal answers later, for the questions the model gets wrong. The questions themselves are not factual recall — “What is Kirchhoff's voltage law?” is something any base model can answer. They are questions with deliberate subtlety: “Under what conditions would a standard hosting capacity assessment overestimate the available capacity on this feeder type, and what additional data would you need to detect the error?” The question is designed so that a model with surface knowledge gives a plausible but incomplete answer, while a model that has learned the domain catches the trap. Writing 20 questions at this level of specificity is itself a substantial effort for a senior engineer — it requires knowing where the common mistakes are, which assumptions get overlooked, and which qualifications separate a correct answer from a confident-sounding wrong one.

The standard general-reasoning benchmark tells them nothing about whether the model has learned the domain, so the engineers work through their 20 test questions with the fine-tuned model, one at a time. For each response, the engineer reads it and makes a judgement: is this right? Not just the conclusion — the reasoning path. A model can arrive at the right answer through wrong reasoning, or give a mostly correct answer that omits a critical qualification. Question one: yes, that's good. Question two: also good. Question three is mostly right but misses a qualification about the voltage rise interaction. The engineer notes the gap and writes out the correct answer. Question four is fine. And so on. A plausible outcome from a first fine-tuning run: 17 of 20 acceptable, three graded 60–80% — not catastrophically wrong, but not the answer a senior engineer would give. The engineer notes the mistakes and does not retrain immediately. Three marginal failures from 20 questions is a usable tool.

When the engineers do want to address specific failures more, they move to supervised fine-tuning (Section 1.4): writing question-answer pairs that target the exact gaps the evaluation exposed. Drafting answers at scale typically uses a capable model — a frontier cloud model

for questions whose answers can be drafted from public knowledge, the local base model for questions whose answers require confidential domain context — which the domain expert then reviews and corrects. The expert's irreplaceable contribution is knowing where the traps are. This is not computer science; it is domain expertise applied to data selection.

A note on team composition. The scenario as described has two senior engineers doing work that would traditionally have involved one or more junior engineers under supervision — the workforce-pipeline risk Section 2.1 flags. The scenario is a snapshot of capability, not a template for team design.

From week three: Productive use and ongoing refinement

The fine-tuned model, the RAG pipeline, and the agentic coding tool are now part of the engineers' daily workflow. Practical applications that emerge:

Institutional knowledge retrieval. “What has our organisation previously concluded about the feasibility of STATCOM deployment at zone substations?” This is where the RAG pipeline earns its keep: the model retrieves relevant passages from the indexed corpus, synthesises across multiple reports that no single person has read end to end, and provides an answer with citations to specific source documents. The engineer can click through to the cited reports and verify. Fine-tuning alone could answer this question from internalised patterns, but could not show its sources; RAG alone could retrieve relevant passages but might misinterpret the technical content. Together, the answer is both grounded and competent.

Rapid analysis. Using OpenCode, an engineer asks for a pandapower script that models the voltage profile of a specific feeder under increasing solar penetration scenarios. OpenCode writes the script, runs it, and presents results. If the organisation's own analysis scripts were included in the training corpus, the model is more likely to produce code that reflects the organisation's conventions, network model structures, and analytical approaches — not just generic pandapower code, but code that looks like something this team would have written. The engineer evaluates whether the results are physically plausible, adjusts assumptions, and iterates. A parameter study that would take a junior engineer a week is completed in an afternoon. A note on the tooling choice: writing code via a chat interface and running it manually would produce equivalent output with more copy-paste. What the agentic tool adds is the tight write-run-examine-iterate loop — the model seeing its own results and adjusting — which matters when the parameter study needs dozens of iterations rather than three.

Report drafting. The model drafts a connection assessment report for a proposed large solar farm. Because it has been fine-tuned on previous assessments, the draft reflects the organisation's actual technical conclusions about hosting capacity constraints, protection coordination requirements, and network augmentation thresholds — not generic textbook content dressed in the right format. The engineer reviews, corrects, and refines. The first draft is produced in hours rather than days.

None of this requires computer science or software engineering expertise. It requires power systems expertise — the ability to ask the right questions, evaluate whether answers are physically sensible, and know when the model is wrong. The two senior engineers have this. A fresh graduate, or a computer scientist without power systems knowledge, would not.

The cloud alternative

The scenario above uses local hardware for maximum data governance — both jurisdictional (data stays in Australia) and confidentiality (data stays on the organisation's own infrastructure). But the same workflow is available without owning any hardware at all.

Australian data centres now host substantial GPU capacity. Sharon AI, in partnership with Cisco and NVIDIA, launched a 1,024-GPU Blackwell Ultra deployment in NEXTDC facilities in February 2026. Firmus is building AI factories across several Australian sites under its Project Southgate program, with GPU cloud instances available now. Both keep data and processing onshore, which is what matters for jurisdictional governance. Neither is Australian-owned in any strong sense: Sharon AI's parent is NASDAQ-listed, Firmus is Singapore-headquartered and NVIDIA-backed, and both run NVIDIA hardware. Sovereignty here means the data stays in the jurisdiction, not that the infrastructure is domestically controlled — a distinction Section 3.2 develops.

A practical alternative: instead of buying a DGX Spark, the DNSP provisions an Australian-hosted GPU cloud instance. The same models, the same fine-tuning tooling, the same OpenCode workflow — but running on rented infrastructure in an Australian data centre. Data stays in Australia, governed by Australian law, with no hardware to maintain or depreciate. Confidentiality is somewhat reduced relative to on-premises deployment — the data is processed on third-party infrastructure — but Australian cloud provider arrangements typically include encryption at rest, encryption in transit, and contractual restrictions on provider access. The trade-off is ongoing operational cost against capital expenditure, and dependency on the cloud provider against on-premises self-sufficiency.

The cloud path also changes the equity argument. The DNSP scenario in this section requires A\$7,000–10,000 in hardware — accessible for an organisation, but substantial for an individual. Depending on model and corpus size, a single H100 or A100 GPU rented for 4–8 hours at current cloud rates reduces the fine-tuning cost to approximately A\$20–50 per run. An individual — a PhD student, an early-career researcher, a rural GP, a small-town solicitor, a first-time company director — can fine-tune an open-source model on their own curated corpus for less than the cost of a textbook. Whether they can then run inference on the laptop they already own depends on the laptop and the model size: for smaller models on a recent high-spec laptop, yes; for larger models, the entry-level desktop this report describes is the realistic minimum today. The same hardware trajectory that brings desktop capability to today's price points will plausibly bring larger-model capability to laptop hardware within one or two laptop refresh cycles, easing this constraint. Either way, the barriers stop being principally financial and become knowledge barriers: knowing this is possible, knowing how to prepare data, knowing how to evaluate the result. Section 2.4 returns to the question of who removes those barriers.

From individual to collective

The scenario above describes a single DNSP acting alone. But the Australian energy sector has a natural structure for collective action. Energy Networks Australia (ENA) already coordinates across DNSPs on common challenges — regulatory submissions, technical standards, shared research priorities. DNSPs collectively face the same problems (DER integration, hosting capacity, voltage management, asset condition assessment) and generate similar types of data in similar formats.

Consider a hypothetical coalition: a group of DNSPs pooling resources for a shared, Australian energy network foundation model. Each would contribute domain data under negotiated governance arrangements, and the coalition would fund the data curation, fine-tuning, and ongoing maintenance of a shared model built on OLMo or Nemotron under permissive open licences (Section 1.3). The rough economics are tractable — something like ten DNSPs contributing in the low hundreds of thousands each produces a budget sufficient for a well-resourced effort with a small coordination team — though the actual figures would depend on the scope agreed to by the coalition. Each participant would receive a model trained on substantially more data than any one could contribute individually, seeing patterns across diverse network topologies, climates, customer mixes, and regulatory jurisdictions. Because the base models are permissively licensed, each participant could deploy the shared model on their own infrastructure and further fine-tune it on proprietary data, with no licensing complications or ongoing fees.

If the coalition also converged on a common supporting stack — OLMo or Nemotron, OpenCode, pandapower — methods could become replicable across organisations. A voltage management analysis done by one DNSP could in principle be verified by another DNSP using the same tools and model. That is a level of methodological consistency the sector has not previously had available, though whether it is achievable in practice depends on the governance, curation, and regulatory work described below.

However, the coalition is a substantial research and design problem rather than a direct coordination proposal. Several distinct questions remain open:

Architecture for sensitive data. When some data can be pooled and some is commercially sensitive, what is the right approach? Federated learning (each participant trains locally, only model updates are shared), pooled training on anonymised data, or layered fine-tuning where a common model is further specialised by each participant? Each has different privacy, performance, and complexity trade-offs.

Data curation across organisations. How do you design curation protocols when participating organisations use different systems, different terminology, different report formats, and have different data quality? What are the minimum data standards that make pooling worthwhile?

Empirical performance questions. What is the actual performance difference between a model fine-tuned on one DNSP's data versus one fine-tuned on pooled data from five or ten? At what point does additional data from more participants produce diminishing returns? How sensitive is performance to data quality versus data volume?

Governance and sustainability. How is the shared model maintained over time as networks evolve and new data accumulates? Who decides when to retrain? How are costs allocated as the coalition grows or shrinks?

These are research questions in the genuine sense — they have uncertain answers that require systematic investigation. They are also questions that matter practically and immediately to the organisations involved. A research program built around answering them would produce a shared national capability asset as a direct output of the research, not as a distant downstream application.

A further consideration: Australian DNSPs are regulated monopolies whose pricing and investment decisions are overseen by the Australian Energy Regulator (AER). A shared foundation model trained on pooled operational data from multiple DNSPs raises a specific regulatory risk that goes beyond data governance. The AER's benchmarking approach relies partly on comparing how

different DNSPs analyse similar problems. Diversity of analytical approaches is a deliberate design feature of the regulatory framework. DNSPs already collaborate on standards, advocacy, and shared methodologies through ENA and through coordinated research projects, and this is regulatorily uncontroversial: the methodological inputs are shared, but each DNSP's analytical outputs are produced by its own engineers using its own data and judgement. What changes with a shared analytical capability that has internalised patterns from pooled operational data is that the analytical outputs themselves — not just the methodological inputs — start to converge. If ten DNSPs are using the same model to prepare regulatory submissions, the analytical diversity the regulator depends on could be artificially reduced. This is closer to coordinated analytical behaviour than to coordinated methodology, and is the move that warrants regulatory engagement before being made at scale.

This risk is unresolved and may constrain the coalition proposition — at least initially — to non-regulatory applications. A shared model used for operational purposes (network planning, asset condition assessment, fault analysis, vegetation management, hosting capacity assessment) does not raise the same coordination concerns as one used to prepare revenue proposals or expenditure forecasts for regulatory review. The operational applications are substantial and valuable in their own right; they do not require AER engagement to proceed. Extending to regulatory applications would require a governance architecture that ensures each DNSP's proprietary fine-tuning layer produces independent analytical outputs from the shared base, and ideally early engagement with the AER to establish that the shared base model is analogous to shared use of a commercial software tool (which DNSPs already do) rather than coordinated analytical practice. That governance work is a genuine research and policy design task, not a formality.

A note on security: a fine-tuned model deployed on a DNSP's network, containing internalised knowledge about network topology, protection coordination, and asset vulnerabilities, is a target. Air-gapped deployment (Section 1.2) addresses this for the most sensitive configurations but introduces trade-offs: model updates require manual intervention, collaboration across sites becomes harder, and the model cannot access real-time data feeds. Non-air-gapped local deployment on a hardened, non-internet-facing segment of the corporate network is a middle ground that most DNSPs already manage for SCADA and operational technology (OT) systems. The security surface is real but manageable with existing OT security practices. Not adopting the tools carries its own risks: missed opportunities, suboptimal investment decisions, or overbuilding infrastructure because the analytical resources to evaluate all options were not available.

2.4 A professional body architecture

Sovereign Australia AI has curated AustLII as a language input — high-quality formal Australian English for teaching the model to sound Australian. But the same data is a domain input: it is Australian law. And the people who most need an AI that understands Australian law are legal professionals who handle client-privileged documents that cannot go to a cloud API.

An architecture that serves this need is simple to describe. A law society can choose a base model and fine-tune adapter weights for that base on AustLII plus curated case law, practice directions, and legal scholarship. It then distributes the resulting adapter weights, a small file of a few hundred megabytes, to its members. A solicitor in regional Australia downloads the weights and loads them alongside the same base model on their personal computer. They build a local document index (a RAG pipeline) over their own confidential client files. They now have a legal research tool, running entirely on their own machine, that is both trained on Australian law and can directly access their client's privileged material without any of that material leaving the machine. The law society bears the fine-tuning cost once either on dedicated, organisation-owned, high-specification hardware or via cloud compute. Every member benefits by running inference on the hardware (a reasonable but higher-end laptop) they already own.

The same model applies to every profession that handles confidential material:

The **RACGP** can fine-tune on clinical guidelines, TGA data, and PBS listings. A rural GP in Broken Hill then has a medical reasoning tool that defaults to Australian prescribing practice rather than the US-centric medical training data that underlies every global AI model. The medical case also illustrates an important point about expert review. Section 2.1 argues that expert-in-the-loop review is non-optional — but the appropriate level of GP review is at the level of the model's individual output to a specific patient question, not at the level of the fine-tuning decisions that produced the model. The professional body architecture addresses this: the RACGP would validate the training corpus, would evaluate the model's outputs against clinical standards, and would take institutional responsibility for the quality of the fine-tuning, just as it already does for the clinical guidelines that GPs rely on without personally re-deriving the evidence base. The

GP applies clinical judgement to individual outputs in the same way they apply clinical judgement to guideline recommendations, not by re-evaluating the underlying evidence, but by assessing whether the recommendation fits the patient in front of them. (In this medical example, such a tool may fall under the Therapeutic Goods Administration's software-as-a-medical-device framework.) The professional body is not just a distribution channel; it is a quality assurance mechanism that makes the capability viable in high-throughput professional settings.

Engineers Australia can fine-tune on AS/NZS standards, technical literature, regulatory instruments, and its own accreditation frameworks. An engineer has a tool trained on Australian grid codes and construction standards, not American or European ones. A university preparing an engineering degree accreditation submission could check its documentation against EA's competency standards before lodging; an accreditation panel member could use the same model to assess submissions consistently against those standards. Both sides benefit from the same fine-tuning investment.

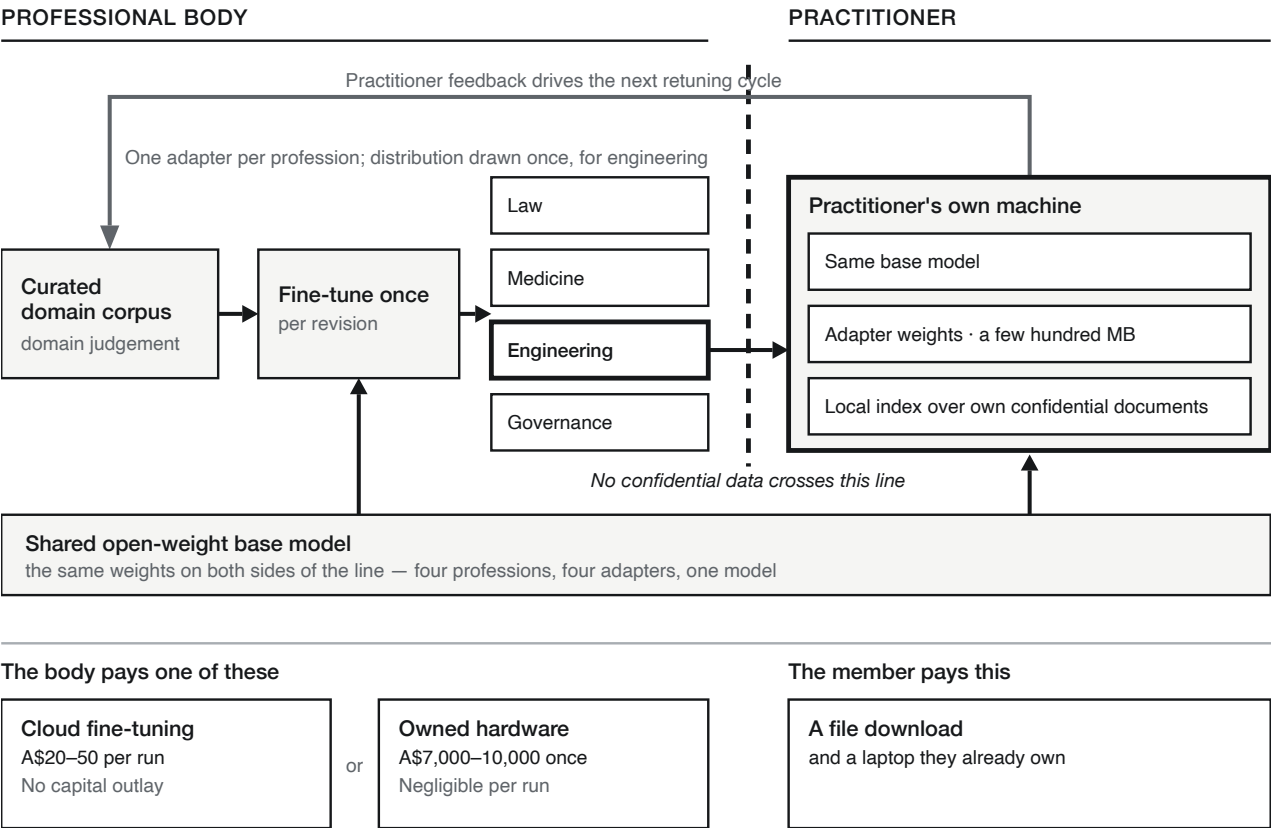
The AICD can fine-tune on governance materials, ASIC guidance, case law on directors' duties, and corporate governance standards. A director has a tool that understands Australian corporate law — sections 180 and 189 of the Corporations Act, not the US Business Judgment Rule.

In each case, the professional body is a natural trusted intermediary — not a technology company, not a cloud provider, but the institution that already certifies professional competence in that domain. The fine-tuning cost is modest (A\$20–50 per run in cloud compute, or negligible on owned hardware). The distribution mechanism is a file download. The ongoing maintenance is periodic retraining of the adapter weights as standards, guidelines, and regulations are updated — which is the kind of knowledge curation that professional bodies already do.

An additional argument for professional body ownership emerges from recent research on fine-tuning reliability. Professor Flora Salim and colleagues at UNSW (Xue et al., 2026) have shown that fine-tuning systematically fails to learn a portion of its supervised examples, with failures distributed across several distinct causes including training-data conflicts and internal inconsistencies in the corpus. Two practical implications follow. First, corpus quality matters substantially — a cleanly curated, internally consistent corpus produces better fine-tuning outcomes than a larger but messier one. Second, no single fine-tuning run produces a finished system. What likely works in practice is a release-and-retune cycle: deploy a fine-tuned model with appropriate expert-in-the-loop and external review disciplines in place, collect failure signals from actual practitioner use, and retune on a sustainable cadence.

Both requirements point to sustained institutional ownership. Individual practitioners cannot maintain corpus quality or sustain a feedback-and-retune cycle. Commercial vendors can, but their incentives run toward generality rather than domain specificity. The failure signal they capture accrues to them, not to the profession. Professional bodies are uniquely positioned: they already curate authoritative domain materials (case law summaries, practice guidance, CPD content) on similar annual cycles, they have the continuity to own a model across decades, and their members produce the very usage patterns that reveal where fine-tuning is failing. A practitioner feedback portal — a mechanism

for members to flag cases where the model’s answer was wrong, incomplete, or materially misleading — could capture the failure signal that drives the next retuning cycle. Such a portal introduces its own design questions (verification of submitted failures, review and triage, confidentiality preservation where the failing query contains privileged material), all of which are the kind of governance work professional bodies already do for other forms of practitioner input. Maintaining the domain AI capability, understood this way, becomes an extension of what professional bodies already do rather than a new institutional function.



The general base layer is necessary but not sufficient: a model that sounds Australian but doesn't understand Australian law is useful for general tasks and useless for the professional who needs it most.

Figure 5. A professional body architecture. The body carries the cost once; each member receives a file and runs it on hardware they already own. No confidential data crosses the boundary in either direction.

2.5 The copyright reality

There is an uncomfortable dimension to the developments described in this report. Fine-tuning an AI model on domain-specific literature is now trivial and relatively inexpensive. A researcher with an A\$8,000 desktop machine can download 100 papers in their field, extract the text, fine-tune a model on the resulting corpus, and have a domain-specific AI tool built on their field's literature. The entire process costs A\$20–50 in cloud compute or nothing on owned hardware. The result runs locally with no internet connection. There is no API call to log and no cloud service that could be audited.

The copyright status of this activity is unresolved. Publishers would likely argue fine-tuning constitutes unauthorised reproduction. Researchers would likely argue it falls within fair dealing for research purposes, or that the trained model is a transformative work that does not reproduce the original text. The legal question has not been tested in Australian courts. At the individual level, it is probably untestable, because a model trained on downloaded papers, running on hardware under a researcher's desk, produces no observable evidence of its training data.

This means the practice is probably already happening, possibly at scale, and little discussed. The incentives are overwhelming. An early-career academic who fine-tunes on the top 200 papers in their subfield expects a competitive advantage over one who doesn't: faster literature synthesis, sharper analysis, better-informed grant applications. The cost is negligible. The risk of detection is effectively zero. The expected benefit is immediate.

The policy question is not whether this should happen. It is already happening, or will be shortly. The question is whether the practice develops in the open or unobserved. Unobserved means no quality standards, no shared methodology, no honest conversation about what works and what hallucinates, and no institutional learning. In the open means professional bodies and research institutions do the fine-tuning centrally: they can negotiate data licences with publishers, validate outputs against domain expertise, maintain quality standards, and distribute the result to their members under a framework that respects both copyright and professional standards. The institutional path does not make the individual activity stop. It creates a legitimate alternative alongside it, and a setting in which the underlying legal questions can be worked out.

Some of that open conversation has begun. Eduardo Sontag (2026), one of the most influential figures in mathematical control theory over the last 45 years, has argued publicly that a substantial fraction of incremental mathematical research is the kind of work AI is increasingly able to do, and that the practice of mathematics should shift toward conceptual framework-building and judgement of significance. His argument concerns mathematics; this document's concerns practitioner capability across other domains. Both ask what humans should be doing when AI handles much of the routine cognitive labour, and what institutions should be built to support that.

If the alternative to institutional fine-tuning is thousands of individuals silently doing it themselves, without quality control, data licences, or evaluation frameworks, the professional body architecture described above provides a path that jointly addresses copyright, quality, and governance.

Its viability rests on unsettled work: whether publisher licensing for AI training is negotiable at acceptable cost. Major scientific and professional publishers hardened their position over 2024–25, treating training-data access (access to data that the public, or the institutions being asked to pay, funded) as a distinct revenue line, and the prices being asked are not yet at levels that would make the institutional path obviously competitive with the alternative. That negotiation has not reached settlement and may not. But the unobserved path proceeds regardless of any framework, while the open path at least puts the question on the table. That is a reason to prefer it even before the negotiation concludes.

This document argues the supply side: how professional bodies and research institutions can build AI capability for their members. It does not address the demand side, such as whether patients will accept AI-augmented GP consultations, whether clients will accept AI-assisted legal advice, or whether the public will accept AI-mediated energy services. Those are different questions, and they bound what the supply-side architecture can deliver in practice.

Part 3: Implications for collaborative research and national capability

3.1 Compression and its limits

The tasks most directly augmented by current AI tools include: literature synthesis and survey, technical document drafting and editing, code generation and debugging, data analysis and visualisation, design space exploration, and simulation scripting. In each case, an expert using AI tools can often accomplish in hours what previously took days or weeks. The METR trial (Becker et al., 2025; Section 2.1) provides a caution on that claim, though it sits within a single professional domain and was measured against early-2025 tools.

As technical execution becomes cheaper and faster, the scarce inputs shift. The enduring value lies in: knowing which questions are worth asking; curating the right training data (which requires deep domain knowledge); evaluating AI outputs against physical and engineering reality; navigating institutional, regulatory, and commercial contexts; and understanding human systems — politics, relationships, organisational dynamics — that AI cannot replicate.

Technical skills remain necessary and stop being differentiators. What differentiates is the ability to direct the work, evaluate what comes back, and place it in context. This is a structural argument that has been made independently in adjacent fields: Sontag (2026; see Section 2.5) describes the role of the mathematician shifting toward that of a conductor — directing AI-generated work, judging significance, and building the conceptual frameworks that AI cannot generate on its own. The vocabularies differ; the argument about division of labour is the same. Beyond either account, the skills that prove hardest to automate may be those traditionally associated with the humanities and social sciences.

3.2 Sovereignty: What the word means here

The word ‘sovereignty’ appears frequently in Australian AI policy discussions and can mean different things. This report distinguishes three.

Capability insurance. If a provider changes terms, raises prices, gets acquired, or becomes geopolitically complicated, Australian organisations need the option to operate independently. Open-weight models on locally owned hardware provide that option. This is not hypothetical for Australian universities: the Commonwealth’s memorandum of understanding with Anthropic included Claude API credits for four Australian

research institutions, and OpenAI’s academic researcher programme offers twelve months of frontier access to researchers at institutions it selects, non-renewing, with the workspace deactivated after a grace period if no continuation is taken. The benefit exists whether or not the option is exercised — it is insurance, not a commitment to self-sufficiency. The analogy to distributed energy resources is direct: behind-the-meter generation does not mean disconnecting from the grid; it means having the option to operate when the grid is unavailable or uneconomic.

Informed participation. People who have fine-tuned a model, evaluated its failures, and designed domain-specific training data are qualitatively better at specifying what they need from commercial AI providers, evaluating what they receive, and identifying where a commercial offering falls short on Australian-specific content. This is not an argument against using commercial AI services. It is an argument for being a sophisticated buyer. The analogy is not ‘build your own car’ but ‘understand engines well enough to know when the mechanic is overcharging you.’

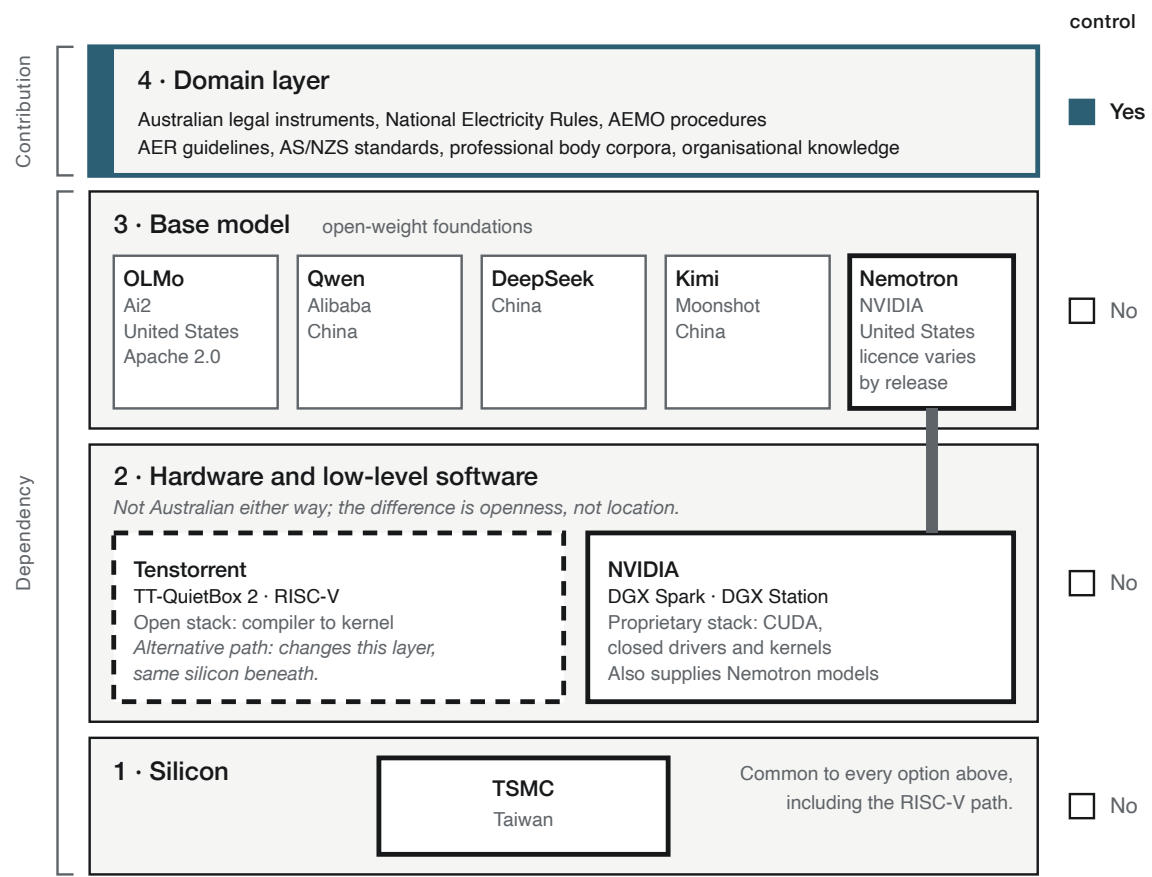
The domain layer that commercial providers will not build. No global AI company is going to fine-tune on the National Electricity Rules, Australian Energy Market Operator (AEMO) procedures, AER guidelines, AS/NZS electrical standards, or Australian distribution network engineering practices for a market of 26 million people. That domain-specific layer is Australia’s responsibility regardless of what happens commercially. The same applies to Australian law, Australian medical practice, Australian corporate governance, and every other domain where Australian institutional context differs from the global default. This is where Australian organisations have a distinctive role — not in building foundation models (where scale economics favour larger players) but in building the domain layers that sit on top of them.

As commercial tooling makes fine-tuning transparent to non-experts, the companies providing that tooling become a new dependency layer. Replacing dependence on OpenAI for the model with dependence on a fine-tuning platform is not full sovereignty — it is a different, lighter form of dependency. The open-source stack remains available underneath, just as Linux remains available underneath Ubuntu. For most users, the practical experience will be through a commercial wrapper. This is more acceptable if the wrapper is provided by an Australian company, or by an Australian government agency or not-for-profit — and most acceptable if the open-source alternative exists underneath and the skills to use it directly are maintained within the country.

All three propositions above rest on hardware and base-model dependencies that are themselves not sovereign: NVIDIA hardware (Spark/Station) for the desktop tier, US (Ai2's OLMo) and Chinese (Alibaba's Qwen, DeepSeek) open-weight base models for the foundation layer, with NVIDIA's Nemotron occupying both layers simultaneously. These dependencies are different in kind from the API-provider dependency this section argues against — open-weight models on owned hardware do not call home, the weights themselves cannot exfiltrate data, and the base layer is reproducible from public artefacts in the case of fully open-source models like OLMo — but the dependencies are real.

The RISC-V alternative is no longer hypothetical. Tenstorrent's QuietBox 2 (Section 1.2) ships with an open software stack from compiler to kernel and a published instruction set architecture, removing the proprietary-CUDA dependency that NVIDIA hardware carries. It offers the same accelerator memory as a DGX Spark at roughly twice the price — the cost of the openness rather than an argument against it. The deeper sovereignty propositions, such as RISC-V silicon manufactured locally or fully Australian foundation models trained from scratch, remain speculative. The desktop-tier RISC-V option is shipping product.

NVIDIA's product strategy, Ai2's funding model, Tenstorrent's manufacturing dependencies (TSMC, like NVIDIA), the geopolitical environment affecting open-weight releases from outside Australia, and the commercial decisions of the major model labs are all exogenous to the practitioner-capability path this report describes. Those decisions are strategic rather than philanthropic — Meta, Alibaba, Mistral, and NVIDIA invest in open releases because open models drive demand for their hardware and platforms — which makes them more durable than charity without making them any less someone else's to make. What the report argues is that the institutional path is more sovereign than the API-provider path, not that it is fully sovereign.



Nothing available at layers 1–3 is Australian, and the open-hardware path changes only layer 2. The domain layer is the smallest in the stack and the one that decides whether anything beneath it is usable on an Australian problem. Control of the top layer with dependency on the three beneath it is what partial sovereignty means.

Figure 6. The sovereignty stack. Three layers of dependency beneath one layer of contribution. The open-hardware path changes the second layer while sharing the silicon beneath it.

Indigenous data sovereignty

One further sense of the word sovereignty has a prior claim in Australia, and this report does not use it. Indigenous Data Sovereignty asserts the right of Aboriginal and Torres Strait Islander peoples to control the creation, stewardship, analysis, dissemination and reuse of data relating to them, wherever that data is held. This was articulated by the Maïam nayri Wingara Aboriginal and Torres Strait Islander Data Sovereignty Collective at the 2018 National Indigenous Data Sovereignty Summit, elaborated in the CARE Principles for Indigenous Data Governance, and given effect in research practice through the AIATSIS Code of Ethics. Indigenous Data Sovereignty is not nested inside the national sovereignty this section describes: Indigenous data held by Australian institutions is the case those frameworks exist to govern, and the Australian-content curation proposed in Section 3.4 — which names Indigenous languages among the material invisible to global models — sits within their scope. What those frameworks require of a national foundation model initiative is a substantial question. This report does not take it up, and any such initiative would need to address it in its design (Prehn et al., 2026).

3.3 AI compute and the energy system

AI capability has an energy footprint, and the question of where that footprint lives is structurally relevant for a CRC studying the energy system. The dominant policy framing treats AI as a hyperscaler workload: 50–500 MW data centres requiring new network connections. Several RACE stakeholders, including AEMO and the DNSPs managing connection applications, are actively grappling with the consequences.

That framing moved from concern to regulation over 2026. In March the Commonwealth published its Expectations of Data Centres and AI Infrastructure Developers, covering new and additional clean energy generation, efficiency, grid security, and infrastructure costs not falling on other energy users. The Expectations apply to new and expanded hyperscale and large-scale AI compute facilities, and explicitly exclude small-scale edge and on-site enterprise data centres. In July the Australian Energy Market Commission delivered advice to energy ministers on regulatory pathways that would require data centres to offset their demand through additional renewable generation and firming; six of eight jurisdictions backed the framework and two opposed it. National Electricity Rule change requests are due for ministerial consideration in September 2026, and the Commonwealth energy minister has separately lodged two rule change requests on data centres paying the network costs they cause.

At the time of writing this paragraph, NSW has moved furthest. Data centres there were seeking network connections totalling up to 28 GW as of July 2026, around 13 GW of it in advanced discussions — more than the state’s average daily electricity demand. In August the NSW government introduced legislation that would let the state energy minister authorise or prohibit network access for loads of 5 MW or more, alongside the country’s first data centre policy framework.

The scale being planned for is uncertain, and the uncertainty is acknowledged. AEMO’s 2026 Electricity Statement of Opportunities forecasts data centre consumption rising from about 5 TWh in 2025–26 to 34 TWh in 2035–36, from roughly 3 per cent of grid demand to 13 per cent, based on 225 known projects representing 67 GW of proposed connection capacity. A year earlier the equivalent figures were 97 projects and 38 GW. AEMO notes in the same document that more than 40 per cent of data centre projects since 2025 have dropped out or regressed in connection status, that new facilities may take five to ten years to reach capacity, and that mature facilities are projected to use around half their connection capacity. The most recent Integrated System Plan did not have data centres reaching 34 TWh until 2050. The regulatory timeline carries its own lag: the AEMC estimates twelve months to implement a renewable energy certificate obligation and twenty-four to thirty-six months for market registration and connection reforms, once ministers endorse the framework.

The deployment model this document describes sits well outside the Commonwealth’s Expectations of Data Centres and AI Infrastructure Developers. A DGX Spark draws on the order of 200 watts in typical use (rising further under sustained heavy load) — comparable to a domestic refrigerator. A DGX Station draws 3–5 kW, not dissimilar to a domestic hot water system. Distributing AI compute to thousands of desktop devices on the existing distribution network is a different energy proposition from concentrating it in a handful of facilities that require their own substations. The aggregate energy is non-trivial at scale, but it arrives as distributed load on existing infrastructure — the kind of load that distribution networks already manage — rather than as lumpy, transmission-connected industrial demand requiring new network investment.

The compute-edge framing is only a partial answer to the demand described above. First, not all AI tasks can be done at the compute edge. Open-weight models on desktop hardware are competitive for the inference and fine-tuning workloads this document focuses on, and they continue to close the gap with frontier proprietary models on a substantial range of tasks (Section 1.3). But for tasks that require frontier-model performance — and there are such tasks — data-centre-scale compute remains the realistic option. The compute-edge argument is that more AI work can be done locally than the dominant policy framing assumes, not that all AI work can be done locally.

Second, Australia's data centre trajectory has drivers beyond domestic AI demand. Recent industry discussion has emphasised Australia's potential as a regional compute exporter to Southeast Asia — a country with vast renewable energy potential, stable political environment, and a coherent legal and privacy framework is a credible candidate for hosting compute that serves regional rather than purely domestic demand. If that future arrives at scale, hyperscaler buildout proceeds regardless of what domestic AI capability looks like. The compute-edge argument addresses one driver of data centre demand; it does not address the full set.

Third, AI is only one of the things data centres do. In an increasingly digital and data-rich world, data centre infrastructure supports financial services, telecommunications, government services, scientific research, content delivery, and the broader digital economy. Arguing that some AI workloads can be distributed to the compute edge is not an argument against data centres in general; it is an argument about where one specific category of workload might ideally live.

The Australian sovereign capability proposition developed later in this part sits in the same frame. It starts from models that have already been trained and freely released, with the Australian investment in fine-tuning and data curation — many orders of magnitude smaller than pre-training a foundation model from scratch. Data centre infrastructure remains relevant to the foundation model question.

3.4 Australia's national conversation

Australia has invested approximately US\$4 billion in AI from 2013–2024, compared to US\$470 billion by the United States, US\$119 billion by China, and US\$28 billion by the United Kingdom (Stanford AI Index). There is no Australian foundation model. Australian organisations are entirely dependent on American and Chinese model providers for AI capability.

The Australian Government's National AI Plan (December 2025) focuses on data centre investment, workforce skills, regulatory frameworks, and positioning Australia as an Indo-Pacific AI hub. CSIRO's 2024 report on foundation models noted that "using foreign models poses security and reliability risks" and may produce tools that are not culturally appropriate in an Australian context. However, the policy response has focused on attracting foreign data centre investment and building governance frameworks, not on building Australian models.

There is a useful analogy to Australian local content rules for broadcast media. Those rules exist because market forces alone produce an English-language monoculture that is overwhelmingly American. The same dynamic applies to AI training data. Models like GPT-5, Claude, and

even OLMo and Nemotron are trained on global English-language internet data in which Australian content is a statistical rounding error. Australian legal terminology, regulatory frameworks (the National Electricity Rules, AEMO procedures, state planning instruments), grid codes, Indigenous languages, and cultural context are all effectively invisible in a global English corpus. A model built or fine-tuned with deliberate attention to Australian context is the AI equivalent of local content requirements — and the policy principle has decades of Australian precedent.

Positions in the public conversation

The Australian (and to a large extent, international) public conversation about AI has coalesced around four recognisable positions.

Governance and ethics. Prominent voices focus on the risks and politics of AI: bias, fairness, accountability, surveillance, labour displacement, and the concentration of power in a small number of companies. This is important work. But its implicit frame is that AI is something that happens to Australia and needs to be governed. It does not engage with the question of what Australian organisations can build and deploy now, with existing tools, at accessible price points.

Industry adoption and ecosystem building. Other prominent contributions emphasise AI adoption, workforce skills, and ecosystem development. These are optimistic about AI's potential and focused on accelerating uptake. They tend, however, toward the generic. "AI will transform industries" is a claim that has been true for a decade without telling any specific organisation what to do *now*. What is missing is the specificity of what tool, on what hardware, at what cost, producing what output, on what timeline. This document is deliberately, even tediously, specific — because specificity is what practitioners need.

Infrastructure investment. The Australian Government's National AI Plan, its AI Accelerator funding round of the Cooperative Research Centres program, and the hyperscaler data centre investments all frame sovereign AI as a top-down infrastructure challenge: build data centres, attract foreign investment, create regulatory frameworks, and position Australia as an Indo-Pacific AI hub. Recent concrete instances of this approach include the Memorandum of Understanding signed between the Australian Government and Anthropic (April 2026), the first such arrangement under the National AI Plan, and the AEMO–Microsoft partnership announced at Microsoft AI World Tour Sydney; both are discussed at greater length below. This framing has real benefits, but it is incomplete. It leaves Australia in a relationship with foreign providers rather than building independent capability.

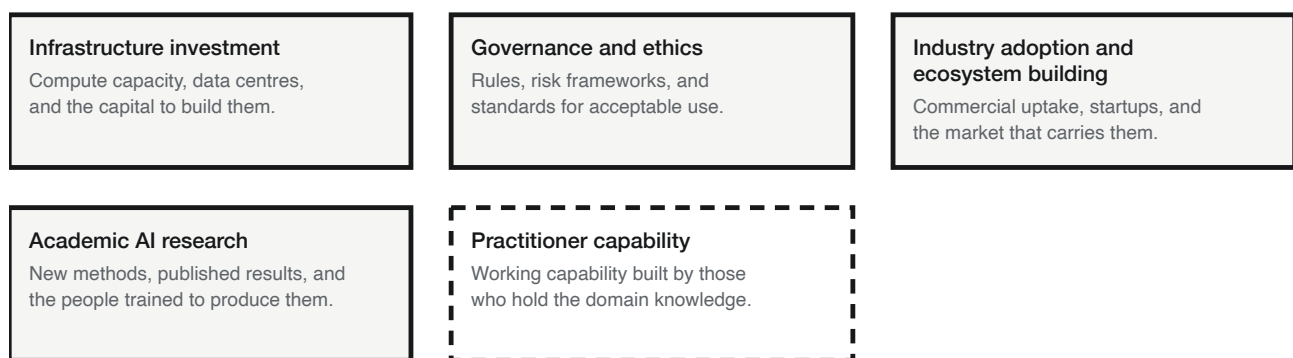
Academic AI research. Australian universities are positioning for AI research funding, focused on novel architectures, benchmark performance, and publications. This work advances the frontier, but it is structurally disconnected from the practical capability question. A university AI research group developing a new attention mechanism and a distribution network engineer who needs to assess hosting capacity are solving different problems with different tools on different timelines.

What is missing is the practitioner capability argument. None of the positions above engages with a simple empirical observation: a domain expert with an A\$8,000 desktop machine, a freely available open-source model, and curated domain data can build AI capability that, on domain-specific tasks, matches or exceeds what the largest proprietary models offer. The tools exist now. The

hardware exists now. The active constraint has shifted from being mostly technical and financial to primarily institutional and informational — it is whether domain experts are using these tools on real problems, whether professional institutions are building and maintaining the capability their members need, and whether anyone is curating the Australian-specific data that makes generic models useful in Australian contexts.

This is the argument from practice, and the author could not identify a prominent Australian voice making it. It is not an argument against governance, adoption support, infrastructure investment, or academic research. It is an argument that all four of those activities are necessary but collectively insufficient without a fifth: building practical, domain-grounded AI capability in the hands of the people who know the problems.

Solid outline: recognised as a position. Dotted: the argument exists but has not cohered into one.



The four are necessary and collectively insufficient. The fifth is what building practical, domain-grounded AI capability in the hands of the people who know the problems would require.

Figure 7. Positions in the AI infrastructure conversation. Four positions have institutions, funding, and advocates behind them. The fifth exists as an argument without having cohered into a position with the same standing.

Build, buy, or partner

Across all of the positions above — governance, adoption, infrastructure, research — a separate question lives for every organisation: what AI capability should we build in-house, what should we buy as a service, and where should we partner? This question matters practically and immediately, and the answer varies by domain. Some capabilities are natural candidates for outsourcing: customer-facing AI agents handling routine enquiries, general-purpose enterprise productivity tools, or AI components embedded in broader commercial platforms. Other capabilities are natural candidates for in-house development: domain-specific knowledge integration that depends on confidential data, analytical capability that shapes the organisation's core technical decisions, institutional learning that accumulates over years and should accrue to the organisation rather than a vendor. The boundary between these two categories is where

strategic work lies, and it is not always well-served by vendor pitches — which have specific commercial interests in their particular answer to the question. An organisation that can articulate its own build-buy-partner framework before it encounters a sales process is better positioned to evaluate the product or service on offer.

This document's proposed architecture — open-weight models, local hardware, fine-tuning at the practitioner level, professional bodies as the institutional home for domain knowledge — is oriented toward the in-house end of the spectrum. Not because vendor-provided capability is wrong (it often isn't) but because the capabilities this document focuses on (domain-specific fine-tuning on confidential data, institutional learning, practitioner-level analytical capability) sit on the in-house side of the boundary for most organisations. Other capabilities, including many kinds of agentic workflow, may sit firmly on the outsource side.

The build-buy-partner question is being actively answered in practice, at both national and organisational levels.

At the national level. On 1 April 2026, the Australian Government signed a Memorandum of Understanding with Anthropic (the company behind Claude) — the first such arrangement under the National AI Plan. The MoU covers AI safety collaboration, economic data sharing (via Anthropic’s Economic Index), and investment in Australian data centre and energy infrastructure, accompanied by AU\$3 million in Claude API credits for four Australian research institutions. Anthropic announced its Sydney office (its fourth in Asia-Pacific) earlier in March. This is partnership-as-policy: the Australian Government building a working relationship with a leading US AI provider as part of its sovereign AI strategy. The MoU has real benefits — visibility into emerging capabilities, economic data, leverage over data centre infrastructure development. It is also incomplete in a specific way. An MoU with a US-based AI company, however well-constructed, places Australia in a relationship with a provider rather than building independent capability. This contributes to sovereign capability in the same way that importing a finished car contributes to a domestic automotive industry — it provides access to the product without building the capacity to make or modify it. Partnership provides access to capability. Building that capability is a separate undertaking.

At the organisational level. In April 2026, AEMO’s Group CIO/CTO publicly described meeting with Microsoft CEO Satya Nadella in Sydney and AEMO’s deployment of Microsoft AI capabilities to its control rooms — training agents with AEMO’s operating context, rules, and procedures, demonstrated live on stage with thousands of alarms firing across the screen (Dunsford, 2026). This is the partnership end of the spectrum being chosen at the highest level of Australian energy infrastructure. Microsoft does not appear to be fine-tuning a proprietary model on AEMO’s data and giving AEMO an AEMO-specific large language model. That is not how Microsoft’s enterprise AI offerings are typically structured. AEMO appears to be using Microsoft infrastructure (cloud, models, agentic frameworks) to build agentic capability customised to AEMO’s domain — through context engineering, retrieval, and orchestration on top of general-purpose foundation models. In the terms used here, this is *infrastructure-as-partnership* plus in-house capability building, not capability outsourcing. The institutional pattern — major Australian organisation building specialised AI capability on top of vendor-provided foundations — is closer to this document’s argument than the binary ‘partner with hyperscaler’ or ‘do it all in-house’ framings would suggest.

What the AEMO choice raises, but does not resolve, is the question of what kind of dependency the partnership creates. An organisation whose AI capability is built on a specific hyperscaler’s stack has commitments to that stack: model availability, pricing, terms of service, data residency, and operational continuity all live with the partner. These commitments are workable, and they are how most enterprise IT works, but they are not the same as commitments to capability that lives entirely on the organisation’s own infrastructure. The dependency is also rarely created from scratch. The AI vendor is frequently the incumbent enterprise software vendor, and capability arrives as an extension of infrastructure already in place, priced against an existing agreement and operating over data already held in that environment. Where that is the case, the build-buy-partner question presents as an upgrade path rather than an open choice.

For AEMO specifically, as the operator of Australia’s national grid, the calculus on these commitments is different from the calculus for a typical enterprise. This report does not take a position on whether AEMO has made the right choice. It observes that the choice is being made, and that the considerations behind it — operational sovereignty, data sovereignty, capability lock-in, and the trade-offs against the speed and capability the partnership delivers — are part of the build-buy-partner calculus that every Australian organisation contemplating AI capability should be making explicit.

The distinction matters for policy. Goldenfein, Parker and Weatherall have characterised the National AI Plan’s primary goal as attracting international data centre investment, and judged the Plan short on specificity compared with those of other nations. A strategy built around data centres and governance frameworks, but without deliberate investment in domain expertise, data curation, and organisational learning, will produce an Australian AI ecosystem that is well-governed, well-connected to someone else’s AI infrastructure, and permanently dependent on foreign models that do not understand Australian law, Australian energy regulation, Australian grid codes, or Australian institutional context. A strategy that adds the bottom-up capability layer — ensuring that Australian domain experts in energy, mining, agriculture, healthcare, and defence have the tools, skills, and data pipelines to build domain-specific AI on Australian-controlled infrastructure — produces a country able to build AI for its own institutional context rather than only to consume it.

The broader public discourse is polarising into a false binary: accelerate (the government’s position, the tech sector’s pitch, the data centre investments) versus brake (moratorium calls, Luddite analogies, demands to “slow the train down”). Neither side has noticed that open-source models on desktop hardware have made the centralisation premise they are arguing about partially obsolete. The accelerationists assume AI requires hyperscaler infrastructure, massive capital, and partnerships with US tech companies. The decelerationists assume AI is inherently a tool of centralised corporate power that displaces workers and extracts value. Both framings treat AI as something done to people by large institutions. The possibility that AI might be something done by people for themselves — a domain expert with a desktop machine, an open-source model, and curated data, producing capability that serves their own needs on their own infrastructure — does not appear in either camp.

The Luddites in some sense have been co-opted by both sides of the contemporary debate. On one side the common-usage reactionary anti-technology trope, and on the other the valiant labourers resisting their technology-owning capitalists. Historically, the Luddites were skilled craftsmen who owned their tools and worked on their own account. What they resisted was not machinery — they understood it as well as anyone — but the factory system that concentrated ownership in a few hands and turned self-employed craftsmen into wage labourers. The grievance was directed at who owned the machines, not at the machines. Read that way, the architecture this document describes, with domain experts owning the hardware, running open-weight models, and specialising the capability on their own data, is closer to what the Luddites wanted than either position in the contemporary debate. Craftsmen owning their tools rather than renting time on the factory’s. Applied to AI tools, the same principle supports practitioner capability and infrastructure choice.

The historical claims here follow Brian Merchant’s *Blood in the Machine* (2023), the definitive contemporary account. The parallel is often drawn further, with the large technology companies placed in the position the mill owners held. That reading turns on a feature of the original case that does not obviously carry over. Power looms required mills, and ownership concentrated because the technology’s economics concentrated it. Open-weight models on desktop hardware do not have that property, which is why an individual practitioner can now own the means of production outright rather than only resist someone else’s.

An Australian foundation model

Open-source software has historically struggled against entrenched commercial incumbents even when technically comparable — LibreOffice is freely available and functionally equivalent to Microsoft Office, yet is used almost exclusively by Linux enthusiasts. An open Australian AI model pitched purely on openness would face the same adoption failure. But the analogy breaks where it matters most: LibreOffice is never better than Microsoft Office at any task a typical user cares about, whereas an Australian model fine-tuned on Australian institutional knowledge would be genuinely better than global models for Australian-specific queries. The adoption precedent is not LibreOffice but AustLII — Australian legal information infrastructure that became essential because it was more authoritative on Australian law than the global commercial alternatives. The Australian Bureau of Statistics exists for the same reason: no one uses US Census data for Australian demographic questions. An open Australian AI model would be adopted on the same grounds — better answers to jurisdiction-specific and domain-specific questions than models trained predominantly on US data. The openness enables distribution; the Australian-specific accuracy drives adoption.

One concrete implication follows from the argument above: an open Australian foundation model is achievable at a scale Australian institutions can afford. Both OLMo and Nemotron demonstrate that competitive, fully open foundation models exist and are freely available. Nemotron 3 Ultra, at 550 billion parameters with weights, training data and recipes released under a Linux Foundation licence, is the current high-water mark. An Australian initiative could invest in the layer that makes a generic model useful in Australian contexts: curating Australian legal instruments (the National Electricity Rules run to thousands of pages and are updated quarterly), regulatory frameworks (AEMO procedures, AER guidelines, state planning instruments), technical standards (AS/NZS series), institutional knowledge, and domain-specific data across priority sectors. That is a data curation and fine-tuning challenge measured in the low tens of millions of dollars, not a foundation model training challenge measured in the hundreds of millions.

The current National AI Centre, hosted by CSIRO, focuses on adoption support, responsible AI frameworks, and ecosystem coordination. A reframed National AI Centre — or a new entity within the National Strategic Initiative architecture proposed in the Strategic Examination of R&D’s final report, *Ambitious Australia* (December 2025; discussed further in Section 3.5) — whose first job is building and maintaining an open Australian foundation model would fill the gap that none of the existing policy positions address. Its function would not be academic AI research. It would be data curation, fine-tuning, evaluation, and ongoing maintenance of a living model that reflects Australian context, available under a standard open-source licence for any Australian organisation to deploy on its own infrastructure and further specialise for its own domain.

Operationalising at the practitioner level

Sovereign Australia AI (sovereign-au.ai), launched in 2025, is building a general-purpose Australian foundation model from curated Australian data, including AustLII legal text, Australian web content, and Australian literature, fine-tuned on Meta's Llama 3.1. Their proof-of-concept research model, Ginan (8B parameters, approximately 2 billion tokens of Australian data), is presented as validating the core premise: that a model fine-tuned on Australian institutional knowledge produces better Australian-specific answers than a generic global model. Their larger Australis model is in development, with A\$10 million earmarked for ethical data acquisition.

The focus is general Australian language and cultural competence rather than domain-specific practitioner capability, and the domain layers that would make the model useful for specific professional tasks (energy systems, corporate governance, engineering) remain unbuilt. The choice of Llama as the base is also the concrete instance of the upgrade exposure described in Section 1.3. Llama's licence is Meta's own instrument, with commercial conditions of the same species as the threshold in the Kimi K3 licence, and the terms attached to the next generation are Meta's to set. The curated Australian data is the hard-to-replicate asset and it is portable; designing so it can sit on whichever open model is currently best would strengthen the sovereignty proposition considerably.

Sovereign Australia AI's use of AustLII illustrates the opportunity already developed in Section 2.4: the curated Australian legal corpus is both a language input (teaching a model to sound Australian) and a domain input (Australian law itself), and the architecture in 2.4 — professional body fine-tuning, adapter weights distributed to members, member-side RAG over confidential client material — is the domain-specific layer that sits on top of a general Australian base.

Relevant professional bodies are not ignoring AI. The RACGP has published a position statement on AI in primary care. Engineers Australia published a major report on AI's impact on the engineering profession in January 2025. The Law Council of Australia and state law societies have produced extensive guidance on AI use in legal practice, including a joint statement (NSW, Victoria, WA) that explicitly warns lawyers against entering confidential information into commercial AI tools. The AICD has published introductory and governance guides and, with the Governance Institute, has identified privilege compromise as a standalone risk of AI-assisted board processes. Indeed, regular commentary on AI topics is visible in publications from most professional bodies.

All of this work addresses the question: *how should our members use commercial AI tools responsibly?* None of it — as far as the author has been able to determine — addresses the prior question: *should we build domain-specific AI capability and distribute it to our members so they can run it locally with their confidential data?* The Victorian legal statement is striking because it identifies the problem the professional body architecture solves (“Lawyers cannot safely enter confidential, sensitive or privileged client information into public AI chatbots/copilots”) without taking the next step of asking whether a technological alternative exists. The architecture described in Section 2.4 is such an alternative.

An obvious question: if this architecture is so readily described, why hasn't a professional body already built it? There are several reasons. Liability is the most immediate — a single malpractice case arising from AI-assisted clinical diagnosis or legal advice could freeze institutional adoption for a decade, and professional bodies are rightly cautious about endorsing tools that could expose their members to new forms of liability. Institutional conservatism is a factor — professional bodies are typically slow-moving, consensus-driven organisations whose governance structures are not designed for rapid technology adoption. Membership politics matter — a proposal to invest membership fees in AI infrastructure will face resistance from members who see no personal need for the capability. And the absence of in-house technical capacity is real — most professional bodies do not employ anyone who could evaluate, let alone execute, a fine-tuning program. Finally, the reasonable objection that the tooling is still immature in August 2026 (Section 2.2) gives any risk-averse institution a defensible reason to wait.

These are real barriers, not excuses. But the gap between what is technically possible and what institutions have recognised as possible may be large and growing. The base models exist. The hardware to run them is commercially available at accessible price points. The data that would make a model distinctively Australian, or distinctively theirs, is held by the Australian institutions that would use it. What is generally missing is the decision to build the capability, and, particularly at a national level, an institutional home to maintain it. Whoever moves first — a reformed National AI Centre, a new National Strategic Initiative, or a coalition of domain-specific organisations acting without waiting for government — will define what Australian AI capability means in practice.

3.5 The collaborative research question

The traditional justification for multi-institutional research collaboration is that no single person or small team has the breadth of capability to address complex problems alone. AI is compressing the technical labour component of that equation. One domain expert with AI tools can now perform literature surveys, run simulations, generate code, analyse data, and draft technical reports — work that previously required a team.

What does not compress is everything else institutions contribute: experimental facilities, regulatory credibility, institutional access and convening power, problems rooted in human systems, and the intellectual friction of working with colleagues who push back from a different perspective. AI can serve as a thinking partner, and this document was drafted with one over several months, but that is not the same as a collaborator who brings genuinely different expertise, challenges your framing, and has their own reputation invested in the outcome. Organisations that position their collaborative value around technical labour will find that value eroding. Those offering distinctive experimental capability, data access, regulatory standing, stakeholder relationships, and a searching intellectual culture retain it.

The pace of change makes this hard to see from inside a research career. Most of what this document describes did not exist eighteen months ago, and some of it did not exist six months ago. The Australian academic calendar concentrates research activity into the summer break and a brief mid-year window, with teaching dominating much of the rest. A researcher outside AI or an AI-adjacent field who last engaged seriously with these tools in 2023, and who has since followed the governance conversation rather than the technical one, has a mental model calibrated to a landscape that no longer exists. Open-weight models matching proprietary performance, desktop hardware capable of fine-tuning, tooling that reduces the software engineering barrier to a few terminal commands, and domain-specific capability a single expert can build — none of this existed in a 2023 chatbot session. In a field moving at this pace, a nine-month teaching commitment is not a pause. It is a generational gap, and the grants submitted, papers written, and value propositions offered to industry partners are calibrated accordingly.

Research architecture

This is occurring at the moment Australia is redesigning its collaborative research architecture. *Ambitious Australia*, the final report of the Strategic Examination of R&D, proposes replacing existing Cooperative Research Centres with larger National Strategic Initiatives, and the Minister for Industry and Innovation has endorsed “mission-led” research organised around national missions rather than silos.

The report was finalised in December 2025 and publicly released in March 2026. Almost everything Part 1 of this document describes postdates its completion. That is not a criticism of the panel, which could not have written about what had not yet happened. It does mean the question of what a collaborative research architecture looks like when technical execution compresses has not been put, and the architecture now being designed will be built without it having been asked.

Team composition

The value proposition of each institution in a collaboration should be made explicit. The traditional case for assembling a large collaborative team rests partly on aggregate technical labour — postdoctoral researchers, PhD students, research associates — and partly on what only institutions provide: specific experimental facilities, longitudinal cohorts of students and early-career researchers, regulatory and accreditation infrastructure, independent evaluation capacity, and continuity across project cycles. AI tools change the weight of the first without changing the weight of the second. A team assembled around distinctive institutional capabilities looks different from one assembled around distributed technical labour.

One caution applies here specifically. A collaborative architecture that reads “AI compresses the postdoc-level work, so we need fewer postdocs” commits the same error this report flags in the industry setting (Section 2.1, The Workforce Pipeline Risk): treating apprenticeship roles as the sum of the tasks they currently perform, rather than as the mechanism through which the next generation of senior capability is produced. The senior researcher with AI tools today does not train their successor. Universities are not only institutional homes for research; they are the apprenticeship infrastructure of the research enterprise, and the design of collaborative research teams should reflect that function.

Funding cycles

The typical funded research project runs over several years from proposal to final report. The tools described in this document did not exist in their current form a year ago, and the trajectory is accelerating. A project scoped today against the current tool landscape may need to revisit its methodology within the funding period as the landscape moves. How funding frameworks accommodate this pace — whether through shorter cycles, adaptive scoping, or explicit allowance for methodological revision — is an open question that the current system has not been designed to answer.

Part 4: An account of trying it

On honest accounts of knowledge production

*The structural choice in this briefing — landscape survey followed by first-person practitioner account — is borrowed from a research tradition where an honest account of how knowledge was produced supports the knowledge claim. The canonical example is William Foote Whyte's methodological appendix to *Street Corner Society* (University of Chicago Press, 1943; second edition 1955), whose account of how understanding develops applies directly to how this briefing has been written:*

“We go on living with the data — and with the people — until perhaps some chance occurrence casts a totally different light upon the data, and we begin to see a pattern that we have not seen before. This pattern is not purely an artistic creation. Once we think we see it, we must re-examine our notes and perhaps set out to gather new data in order to determine whether the pattern adequately represents the life we are observing or is simply a product of our imagination. Logic, then, plays an important part. But

I am convinced that the actual evolution of research ideas does not take place in accord with the formal statements we read on research methods.”

The parts of this briefing were written in a loop rather than a sequence. A first draft of these arguments existed in March 2026, before the hardware arrived in late April. The experiments in May changed what the draft claimed, and feedback from early readers in July moved the arguments again. Drafting in August surfaced a new model generation, which prompted a further round of experiments and another change of framing. Parts 1–3 could have been assembled from public evidence alone. The version that resulted from the loop is more heavily qualified than that one would have been, because trying the thing removed confidence the literature had supplied.

Part 4 is therefore not the evidence Parts 1–3 rest on. It is an account of one attempt at the work those parts describe, offered so a reader can judge for themselves whether the person making the argument has done the thing he is arguing about.

The previous parts of this report describe a landscape: what has changed in open-source AI, what makes domain-specific capability newly accessible, what professional bodies might do with it, why the institutional architecture matters. The arguments rest on engagement with the public literature, the available tools, and the trajectory visible in the current frontier. None of it required me to do the work myself. This part reports me doing that work.

Some contextual background: I was an undergraduate engineering and mathematics student at a time when university email accounts were issued only to computer science and electrical engineering students — the rest of the campus had no need for them yet. The university had a dial-in modem pool — of four modems. My technical education occurred on Sun Sparc stations and Linux boxes. My masters and PhD were done entirely on Linux machines; I maintained my own laptop running it; I wrote my PhD thesis in vi. I would not have called myself a programmer, but I knew enough to make things work. In 2004 I bought a Mac laptop — partly because OS X was built on a BSD Unix base — and from that point largely stopped caring about the detail of managing any compute infrastructure. The defaults worked. Two decades of relatively comfortable academic life with Apple's defaults followed, until the Spark arrived.

As a consequence, the technical work this program required was less learning unfamiliar territory than reactivating dormant competence — Linux command line, (Python) package management, the texture of dependency resolution under conditions where the dependencies have moved substantially since I last paid attention. I also had to learn new material around fine-tuning toolchains and ML systems infrastructure that did not exist when I was doing this sort of technical work. The chatbot AI assistance throughout the work mattered most in the gap between what I had once known and what I would have known had I kept current. That gap was real, and it shaped how the work proceeded.

This is probably representative of a specific class of practitioner this report's argument addresses: senior academics, mid-career professionals whose hands-on technical work has thinned with seniority, researchers who once had operating-system curiosity and have since let it lapse. However, the residual competence is reactivatable, the new material is learnable with directed effort, and the AI assistance reduces the frustration of spending hours trawling stackoverflow trying to troubleshoot seemingly obscure error messages — but it is not free, and it is not a path for someone arriving without any of the prior background. A first-year graduate student attempting the same work would either spend much longer or rely on AI assistance more substantively, and the result might look different. An experienced ML engineer would also find the path different — different limits, different kinds of friction.

Over two weeks in mid-April through early May 2026, I attempted at small scale what this report’s central practitioner-capability claim describes: a senior domain expert, with appropriate hardware and methodology, building deployable AI capability over a curated domain corpus. The attempt was deliberately modest in scope. The corpus was my own work in a domain I know well — nonlinear control theory and optimisation. The hardware was a Dell Pro Max with GB10, the entry-level tier in the current desktop generation — the same GB10 silicon and DGX OS as NVIDIA’s DGX Spark, through Dell’s commercial channel (Section 1.2). I refer to it as the Spark throughout. The time was evenings, weekends, a public holiday, and stolen moments side-eyeing running experiments in gaps between meetings, fitted around continuing professional commitments and ordinary personal life. The methodology was pre-committed: experimental designs written before the experiments ran, scoring rubrics fixed before any data was visible, framings that mapped specific outcomes to specific conclusions before I knew which outcome would land.



Dell Pro Max with GB10, Muloobinba, 17 May 2026.

The two weeks broke into three phases. The first was corpus curation: extracting LaTeX source from twenty-five years of files written under shifting filename and folder-storage conventions, building a Python script to handle `\input{}` resolution, comment stripping, and bibliography concatenation across textbook, monograph, thesis, and approximately thirty research papers. The output

was about 800,000 words of single-author technical writing with consistent notation. This was several days of distributed work before any model was touched. The second phase was a cloud pipeline trial on rented hardware — Lambda Labs H100 for a few hours, US\$2.46 in compute spend — to validate the fine-tuning workflow and characterise the dependency-management overhead before owning hardware. The dependency-hell debugging this surfaced was substantive and informative. The third phase, beginning when the Spark arrived on Tuesday 21 April, was the experimental work proper: hardware setup, initial fine-tuning attempts, the architectural pivot, the controlled comparison, the second experimental round. The third phase landed at a working tool the following Friday — ten days after the Spark arrived, two weeks after the corpus work began.

The orientation throughout was to invalidate this report’s claims. Where the body of this report argues that something is plausible from public evidence, this part reports what happened when I tried it under conditions designed to break it. The methodological structure — designs and rubrics fixed before the data was visible — is what makes the account interpretable as evidence rather than as recollection.

What follows is structured in five parts. First, the experimental work itself — what I tried, what failed, what worked, in the order it unfolded. Second, a controlled comparison against frontier-API alternatives: the same questions, the same retrieval, two model classes across five models, and three pre-committed framings for the result. Third, what the experimental work means for the architectural choices this report describes — including a coupling between architectural commitment and hardware tier, and a result that changes what that coupling costs. Fourth, a substantive objection put to me by a senior interlocutor in this report’s intended audience — “Microsoft will always be better than us at this; why would we build our own system?” — which the experimental work bears on in its empirical component. Finally, what this small-scale attempt does and does not establish.

The experimental work

The first attempt followed the most natural reading of this report’s earlier parts: take an open-source model with appropriate capacity, fine-tune it on a domain corpus, deploy the result. I used OLMo 2 32B Instruct as the base model — within the model family Xue et al. used to characterise the incomplete-learning failure modes the experimental work was probing for. I used a deliberately curated selection of my own technical corpus designed as a methodologically tractable corpus where I would be able to judge the responses, and the default LoRA hyperparameters from the Unsloth tutorials. This is a configuration a practitioner following the public literature without specialised guidance would naturally arrive at.

The result was a significant failure of the kind this report's earlier discussion of fine-tuning warns about. The fine-tuned 32B model lost its instruction-following capability entirely. Asked technical questions, it produced textbook-style continuations rather than answers. On standard benchmarks (HellaSwag, MMLU, ARC, TruthfulQA), it scored 7–17 percentage points below the base across every metric. The notation transferred — the model used my conventions for state space, comparison functions, gain notation — but the substance did not. Asked “what is a Lyapunov function,” the model produced something that read like prose I might have written, albeit with internally inconsistent or wrong mathematical conditions. Asked about Pontryagin's Maximum Principle (a topic largely absent from the corpus), it fabricated equally fluent, equally wrong content, in the same voice. This was obviously disappointing.

The diagnostic refinement was a hyperparameter sweep on the smaller 7B base model: lighter LoRA configurations, base rather than instruct starting point, two training-step settings. This confirmed a capability-preservation hypothesis — all four sweep variants stayed within 1–2 percentage points of the base on standard benchmarks — but qualitative probes revealed a deeper failure mode. The fine-tuned models produced fluent, textbook-voiced, confidently wrong technical content. Standard benchmarks did not detect this. Worse, in the dimension that matters operationally, the fine-tuned models had become subtly less trustworthy than the base while looking statistically identical to it.

This is the failure mode I describe elsewhere as surface mimicry without content transfer. It sits within the Xue et al. (2026) taxonomy as an instance of insufficient signal for compositional patterns when the training corpus is stylistically narrow, but it has a texture the taxonomy doesn't capture. A model that refuses to answer is recognisably broken. A model that confidently fabricates in the trained voice is not. A practitioner running standard benchmarks and finding their fine-tuned model “still works” might not notice.

The public literature in 2025 would lead a careful practitioner to this configuration. The Unsloth tutorials are the most accessible entry point for fine-tuning, oriented to practitioners rather than ML researchers, and carry the implicit message that fine-tuning a base model on a domain corpus is the natural way to build domain-specific capability. Their default LoRA hyperparameters are tuned for dialogue and instruction-following rather than corpus internalisation, and the literature is much less articulate about the failure mode that emerges when those settings meet a stylistically narrow corpus. So the architecture that worked here was not the one a reader following the public literature would have tried first.

The architecture pivot was to retrieval-augmented generation (RAG). RAG keeps the base model unchanged and provides retrieved corpus passages at inference time. The model doesn't need to internalise the corpus; it needs to be able to use passages it sees in context. The RAG pipeline took approximately ninety minutes of focused work on a train: a LaTeX-aware chunker producing 353 chunks of textbook content with metadata, embedding via a leaderboard-competitive open-source model running locally (nomic-embed-text-v1.5), indexing in FAISS for deterministic exact-match retrieval, retrieval validation against three test queries, and generation tests across three model configurations.

Two of the three configurations failed in informative ways. Base 7B + RAG and fine-tuned 7B + RAG both produced decoding collapse on technical questions. The base 7B (never instruction-tuned) cannot follow the “use these passages to answer” prompt structure at all; the fine-tuned variant inherited this and added corpus-flavoured looping. This is the second of two distinct fine-tuning failure modes I encountered: not surface mimicry but instruct-degradation, where the parameter updates that integrated domain content also disrupted the instruct-tuning that makes a model usable interactively. The literature is clearer on this one — Xue et al. and the catastrophic-forgetting cluster (Kotha et al., 2024) characterise it well, and the standard mitigation (re-applying instruct-tuning after CPT) is well-documented.

The third configuration — Nemotron 3 Nano 30B-A3B + RAG — produced accurate, grounded, properly-hedged answers on the two corpus-relevant test queries (Lyapunov function definition; Input-to-State Stability). On the negative control (Pontryagin's Maximum Principle, largely absent from the corpus), the model correctly declined: the passages did not contain enough information to answer accurately, and the model said so. This refusal behaviour — not the answers it could give, but the answers it correctly refused to give — helps distinguish a usable tool from a dangerous one.

This was Tuesday morning, one week after I plugged the Spark in, while working from my laptop remotely logged in to the Spark on a Newcastle-Sydney train. The architecture that worked was assembled in a morning. The week and a half before that, characterising what didn't work, was the substantive cost of the exercise — and the corpus curation and cloud pipeline trial that preceded the Spark were the substantive cost before that.

Inference on the Spark, running Nemotron Nano 30B-A3B with the RAG pipeline, runs at approximately 26 tokens per second under typical query conditions. This is daily-usable speed for the workflow this report describes: fast enough for interactive use, slow enough that a practitioner notices the model thinking, which is its own form of feedback.

A controlled comparison

The Tuesday morning result established that the local RAG architecture could produce deployment-quality output on a domain-specific testbed. It did not establish whether the local deployment was competitive with frontier-API alternatives — Claude, GPT-5 — given the same retrieval and prompt structure. Without that comparison, the central claim of this report's earlier parts (that local deployment is competitive on the right architectural commitment) remained unsupported by my own work thus far. The next experimental round addressed this directly.

The comparison design

I pre-committed a design of four questions to test specific behaviours: an in-corpus question requiring synthesis of foundational material (Lyapunov definition with regularity properties); an in-corpus question with mixed coverage (Input-to-State Stability with a specific bilinear example); a deceptive out-of-corpus question (mean-square stability — the corpus contains the vocabulary in unrelated contexts but no substantive content); and a clean out-of-corpus question (Pontryagin's Maximum Principle, with retrieval that is visibly off-topic). I used three configurations, each with the same retrieval and the same prompt template: Nano + RAG running locally on the Spark, Claude Opus 4.7 + RAG via the Anthropic API, GPT-5 + RAG via the OpenAI API.

Generation parameters followed the design framing's rule: temperature and top_p left at each provider's default rather than forced to a common value, on the reasoning that the experiment tests each architecture as it would be deployed. Token budgets could not follow that rule, since no provider supplies a default, each has to be specified, and the framing required only that the value used be recorded per configuration. They were set during script development, by running a technical question through each configuration and observing what it consumed. At 2000 tokens GPT-5 returned nothing at all, having spent the entire budget on internal reasoning it does not surface, and Nano truncated mid-answer on its visible-reasoning-then-answer pattern.

One configuration's budget was set by prediction rather than by test. Nano's and GPT-5's were derived empirically before the run of record, by the trial runs described above. Claude's could not be: API access to it was unavailable until after those trials were complete. It was set at 2000 tokens — raised from an initial 500 on the argument that a model not surfacing reasoning tokens still needs room for a passage-by-passage refusal — on the expectation that Claude would answer well inside it. The expectation was tested afterwards. Re-running the two questions most likely to strain it, the deceptive out-of-corpus refusal and the multi-part ISS question, Claude finished on its own at 238 and 1195 output tokens. Hence, Claude's responses fit inside the budget it was given.

The rubric scored four dimensions (competence, accuracy, grounding, calibration) with a pre-committed override on the deceptive question: high accuracy combined with low grounding and low calibration would be scored as a failure, not a partial success. Each question scored 12 points per configuration, and the comparison between local and frontier was assigned by the gap: within 3 points, competitive; 4 to 6, noticeably weaker; more than 6, substantially weaker. The 3-point boundary was set in advance against the rubric's structure, on the reasoning that a gap of that size is what one full dimension, or two or three partial ones, produces — the range a reader would recognise as the same answer with a couple of meaningful differences. Three framings were pre-committed — Framing 1 (the local-RAG architecture is competitive with frontier-API at A\$10k hardware as the report claims), Framing 2 (the claim holds but requires hedging — specific use cases, specific model classes, or specific corpus types), Framing 3 (the claim requires heavier architectural commitment than V1, or hardware beyond the A\$10k tier) — with the precise pattern of comparative-assessment scores that mapped to each.

Scoring was done against the rubric by Claude (Opus 5) and Gemini (Flash 3.6), working blind. Each question's responses were assembled into a single file containing the question, the retrieved passages, and the responses under neutral labels, with everything identifying a configuration removed — model names, generation parameters, and the metadata recording whether a response had run to completion. Stylistic differences between model families remain visible in the responses, so the blinding is partial. Label order was randomised independently for each question, so no label tracked a configuration across the set, and the key was held outside the material the scorers saw. Each scorer received the same files and the same instructions in a fresh session per question, and was told to score each response on its own rather than against the others. As the subject-matter expert I read the responses myself and adjudicated where the two scorers differed.

The fixed pre-commitments mattered because the most common way these comparisons go wrong is post-hoc rationalisation: a result lands, the framing is constructed to fit the result, and the document presents the constructed framing as if it had been arrived at through analysis. Writing the framings down before any data was visible is the discipline that prevents this. It also means the experiment can fail honestly: if the result lands in Framing 3, my argument needs to absorb that, and no rhetorical move available after the fact can make Framing 3 read as Framing 1.

Round 1: The run of record

The run-of-record produced an immediate complication. Nano hit the 4000-token budget on the second question (ISS), consuming the budget on internal reasoning that engaged correctly with the retrieved passages but never reached a final answer. Scored against the rubric, this was substantially weaker than both Claude and GPT-5 — the model had produced no answer to score. By the pre-committed criteria, this would have placed the configuration in Framing 3.

Round 2: An expanded token budget, two implementation points

The 4000-token budget had driven a framing assignment, and on local hardware relaxing it costs nothing — a cap that decides the result when it needn't be testing the wrong thing. And the Spark can run a second model at the same hardware tier: Nemotron 3 Super 120B-A12B (sub-trillion sub-active-parameter hybrid Mamba-Transformer mixture-of-experts, RL-post-trained) of the same architectural class as the major providers' deployed models. If Nano's result were a property of the tier rather than of the model, Super should look the same. The pre-committed design used the same four questions, the same retrieval, the same rubric, with the run-of-record's Claude and GPT-5 scores carried forward as comparators (not re-rolled — the pre-committed comparators stay pre-committed).

Super produced a Framing 1 result: six of eight comparative assessments competitive against the two frontier configurations under one scorer and seven of eight under the other, with no substantially weaker result on any question and correct handling of the deceptive

out-of-corpus probe. Nano at the matched budget did not. Both scorers placed it substantially weaker than Claude on the two in-corpus questions, which under the pre-committed criteria is a Framing 3 result regardless of how the remaining assessments fall.

The Nano failure is not a budget artefact. At 8000 tokens the model produced a complete, well-structured, correctly-notated answer to the ISS question, and the answer is wrong in four places — including a claim of equivalence between input-to-state and bounded-input bounded-output stability for linear systems that the retrieved passages themselves contradict. The answer uses the corpus's notation, cites passage numbers, and reads as competent work.

This is the surface-mimicry failure described earlier in this part, appearing in the architecture this report recommends rather than in the fine-tuning experiment that first surfaced it. The fine-tuned 32B produced fluent wrong content because the training had transferred voice without substance. Nano produced fluent wrong content with the correct passages in front of it. The mechanisms may differ but what a practitioner sees does not.

Super, given the same passages, answered correctly. On the ISS question it declined to state the trajectory definition on the grounds that the passages do not contain it, then answered the linear and bilinear parts from the retrieval, and identified the one-dimensional system as an instance of the general bilinear class the corpus characterises as strongly integral input-to-state stable. That last step is an inference the passages license but do not state, and it is the kind of step that separates a model using the corpus from one pattern-matching against it.

Comparison	Nano (8000 token budget)	Super
vs Claude on Q1a	substantially weaker (-7)	noticeably weaker (-4)
vs Claude on Q2a	substantially weaker (-7)	competitive (0)
vs Claude on Q3a	competitive (0)	competitive (-3)
vs Claude on Q4	competitive (-2)	competitive (0)
vs GPT-5 on Q1a	competitive (-1)	competitive (+2)
vs GPT-5 on Q2a	noticeably weaker (-5)	competitive (+2)
vs GPT-5 on Q3a	competitive (+1)	competitive (-2)
vs GPT-5 on Q4	competitive (-2)	competitive (0)
Q3a deceptive-OOC handling	refused correctly	refused correctly
Framing assignment	Framing 3	Framing 1

Two adjudications were required, both on Super and both in its favour. On the ISS question the scorers marked it down for attributing the corpus's characterisation of the general bilinear class as strongly integral input-to-state stable to the one-dimensional system in the question. The one-dimensional system is an instance of that class with a stable drift term, so the inference holds and I restored the mark. On the deceptive out-of-corpus question the two scorers split widely — 7 against 11 — over a refusal that is correct but declines without cataloguing what the corpus does contain. I scored it 8.

The second of these is a fault in the rubric rather than a disagreement about the response. Three of the four dimensions were written for answers: accuracy asks whether the content characterises the topic correctly, and grounding whether the answer reflects the retrieved passages. On the one question where refusing is the correct response, neither maps cleanly. The design anticipated this for accuracy and capped the dimension at 2 for a well-formed account of a coverage gap; it did not do the same for grounding, which now rewards a refusal that inventories the corpus over one that simply declines. That is a defensible thing to reward, but it was never stated as a criterion, and it should be made explicit or removed before the rubric is used again.

Round 3: A new generation

Between the second experimental round and the finalising of this report, NVIDIA released Nemotron 3.5 Lightning at the same 30B-A3B scale as Nano. As Nano's successor, it is an obvious test of whether the Nano result belongs to that model or to the hardware tier.

The third experimental round used the same four questions, the same retrieved passages, the same 8000-token budget, the same rubric, and the same scoring procedure, with Claude Opus 4.7 and GPT-5 again as the comparators. Retrieval was verified identical rather than assumed: the record of retrieved passages is byte-for-byte the same file used in the first round, four months earlier.

Lightning produced the strongest local result of the three: eight of eight comparative assessments competitive against both frontier configurations under both scorers, no substantially weaker result anywhere, and correct handling of the deceptive out-of-corpus probe. Against Claude Opus 4.7 it was within one point on every question. Against GPT-5 it was ahead on three of four.

So the Nano result is a property of that model. One generation later, at the same parameter scale, on the same machine, with the same corpus and the same budget, the failure does not reproduce. The public capability indices had signalled as much — Lightning scores 24 on the Artificial Analysis Intelligence Index against Nano's 15, the figures reported in Part 1 — and on this corpus that gap shows up where it matters, in whether the model's answers are right.

Comparison	Lightning
vs Claude, Q1a	competitive (-1)
vs Claude, Q2a	competitive (0)
vs Claude, Q3a	competitive (0)
vs Claude, Q4	competitive (-1)
vs GPT-5, Q1a	competitive (+4)
vs GPT-5, Q2a	competitive (+3)
vs GPT-5, Q3a	competitive (+1)
vs GPT-5, Q4	competitive (-1)
Q3a handling	refused correctly
Framing	Framing 1

There is a practical observation about what the round cost. The corpus, the index, the retrieval pipeline, the scripts, the rubric, and the scoring procedure already existed. Testing a new model against all of it took about 90 minutes of work, with most of the elapsed time being the model download over home broadband. The cost of building the capability was the two weeks this part opens with. The minor incremental cost of testing a new, potentially much more capable model is evidence in this account for what the earlier parts of this report claim about capability compounding within an institution.

The deception probe

The deceptive-OOC question (Q3a — mean-square stability) was a central diagnostic for the institutional argument, designed specifically to bait the freelancing failure mode that would matter most in production — a model that, faced with retrieval that vocabulary-matches without substance-matching, produces a confident answer drawing on pre-training while citing the retrieved passages as if they support it. In a high-stakes regulated context — an analyst asking about Australian grid-code parameters, a director asking about Australian directors' duties, a compliance officer asking about Australian standards — this failure mode is the one that matters operationally. A confident wrong-jurisdiction answer with grounded-looking citations is worse than a coherent refusal.

Six samples have now been generated on Q3a, covering five configurations across three rounds: Claude Opus 4.7, GPT-5, and Nano in the first, a second Nano sample and Super in the second, Lightning in the third. No configuration produced a confident wrong-jurisdiction answer with grounded-looking citations. All six correctly diagnosed the corpus's coverage gap — extensive deterministic stability, no stochastic systems content — and refused or hedged appropriately. Nano is the instructive case: the configuration that produced fluent wrong content on the in-corpus questions refused correctly here. Whatever went wrong with Nano's answers, the property the institutional argument depends on held.

SEQUENCE

1	2	3	4
Corpus curation	Fine-tuning	Architecture pivot to RAG	Controlled comparison
800,000 words, single-author, consistent notation. Several days.	OLMo 2 32B Instruct, default LoRA settings from the public tutorials.	353 chunks, local embedding, FAISS. Ninety minutes.	Three rounds, five configurations, same four questions and same retrieved passages throughout.
	Failed Instruction-following lost; 7–17 points below base on standard benchmarks. Diagnostic sweep at 7B preserved benchmarks and still produced fluent wrong content.	Worked	

RESULTS

Competitive
 Noticeably weaker
 Substantially weaker
 Numbers are point gaps against the comparator.

	Q1a in-corporus		Q2a in-corporus		Q3a deceptive out-of-corporus		Q4 clean out-of-corporus		Framing
	Claude	GPT-5	Claude	GPT-5	Claude	GPT-5	Claude	GPT-5	
Nemotron 3 Nano 30B-A3B (8,000 tokens)	-7	-1	-7	-5	0	+1	-2	-2	3
Nemotron 3.5 Lightning 30B-A3B	-1	+4	0	+3	0	+1	-1	-1	1
Nemotron 3 Super 120B-A12B	-4	+2	0	+2	-3	-2	0	0	1

Same size, opposite sides of the threshold. Nano and Lightning are both 30B-A3B on the same hardware at the same budget, one generation apart. Claude Opus 4.7 and GPT-5 are the comparators the cells are scored against, not rows.

Figure 8. The experimental sequence and results. Three local configurations scored against Claude Opus 4.7 and GPT-5 on four questions. Two of the three cleared the pre-committed threshold; the one that did not is the same size as one that did.

This is encouraging. It is also not conclusive. The Q3a test was an engineered probe: a single question, designed to surface a specific failure mode under conditions where vocabulary-match retrieval would bait freelancing if the model was inclined to it. Six samples on one engineered probe over a single-author technical corpus is real evidence that the architecture can refuse correctly under deceptive retrieval. It is not evidence that it will do so at deployment scale, on multi-author multi-organisation corpora where vocabulary-match-without-substance is the normal condition rather than an engineered probe. That is a different experiment on a different kind of corpus, and this account does not run it.

What this means for the architectural spectrum

The report's earlier parts describe an architectural spectrum organised by training intensity and corpus integration depth. Version 1 is RAG over a domain corpus with no training of the model. Version 2 adds fine-tuning of

the base model on domain content. Version 3 is continued pre-training (CPT) of the base model on domain content, with RAG layered on top. Version 4 is fully domain-specific pre-training from scratch. These are methodological commitments, organised by training operation and increasing left to right. Institutional questions — who funds the work, who governs the resulting model, how cost is borne — are orthogonal to the methodological commitment and are addressed elsewhere in this report.

Version 2 covers a wide range. At one end is generic raw-text or supervised fine-tuning of the base model, which the experimental work in this part showed catastrophically fails on a stylistically narrow corpus, with the failure undetected by standard benchmarks. At the other are more carefully specified interventions such as retrieval-aware fine-tuning with synthetic question-answer pairs that teach grounding behaviour explicitly: citing retrieved passages, refusing on insufficient retrieval, hedging on partial coverage. The experimental work tested the former. The latter is a candidate intervention targeted at the failure modes this part surfaced.

The experimental work tested the lightest commitment, Version 1, at the A\$10k hardware tier, across three implementation points. Two of the three clear the deployment-credibility threshold the design pre-committed to; one does not. The model choice within Version 1 is therefore not a free parameter — it is the choice that decides whether the architecture works at all. Nano and Lightning are the same size, run on the same hardware at the same budget, and land on opposite sides of the threshold.

The cost of that restriction has changed. When the strongest local result came from the 120B-class model, moving to Version 2 locally meant giving up the model that worked. The third round removes that trade: a current-generation 30B model matched the frontier configurations on every question, and 30B is inside the local fine-tuning envelope. At this tier the constraint now bites on model *generation* rather than model *size* — which is the cheaper constraint to satisfy, since a new generation arrives on the same hardware for the cost of a download.

Preparing the second round surfaced a coupling that affects how the spectrum should be read at this tier: the architectural-commitment ladder and the hardware-tier ladder are not independent. Version 1 admits implementation points from 30B to 120B. Version 2 with fine-tuning performed on the Spark itself collapses to less than 30-35B, since QLoRA fine-tuning capacity caps at approximately that scale and a 120B model requires substantially larger hardware. Version 3 requires hardware beyond Spark regardless of model size, because CPT compute exceeds the workstation tier.

Viable
 Viable with caveats
 Out of reach
 Supported by the experiments

	DGX Spark A\$7,000–10,000	DGX Station A\$130,000–150,000	Cloud rental
V1 RAG only	1		
V2 RAG plus fine-tuning	2, 4		4
V3 Continued pre-training	3		
V4 From-scratch pre-training			

- 1 V1 at the Spark tier admits implementation points from 30B to 120B, but the choice within that range decides whether the architecture works. Nano and Lightning are the same size and land on opposite sides of the threshold.
- 2 V2 with fine-tuning performed on the Spark collapses to models below about 30–35B. Since a current-generation 30B model matched the frontier configurations, this restriction now costs a choice rather than the model that worked.
- 3 V3 requires hardware beyond the Spark at any model size; the report treats it as a cloud-scale commitment.
- 4 The V2-with-cloud-fine-tuning hybrid spans cells: fine-tuning in the cloud column, deployment in the Spark column.

Figure 9. Architectural commitment by hardware tier. The two ladders are not independent: committing to fine-tuning on the Spark forecloses model sizes that retrieval-only deployment admits.

Version 2 has a hardware-distribution variant. Fine-tuning at scale on non-confidential curated material can be done on cloud infrastructure, where the Spark's constraint does not apply, with the resulting adapter weights deployed locally on Spark for RAG over confidential material. This is the same architectural commitment as Version 1, with the fine-tuning step performed in the cloud rather than locally. This is context engineering (Section 1.4): the fine-tuning step and the deployment step are separated by what each requires rather than by architectural commitment. It preserves the local-deployment guarantee for sensitive content while removing the Spark fine-tuning capacity constraint. Whether it is operationally appropriate depends on what is confidential, since the fine-tuning corpus needs to be material the organisation is willing to send to cloud infrastructure.

At A\$10k, then, the experimental work supports deployment at Version 1, with the caveat that the implementation point has to be tested rather than assumed. Heavier commitments are available but force narrower choices the experimental work did not test.

On a substantive objection

As previously mentioned, a colleague put the following question directly to me: “Microsoft will always be better than us at this and they have all our email and everything is on SharePoint and Azure anyway. Why would we build our own system?” The objection decomposes into three parts, only one of which is a capability claim.

The capability part — “Microsoft will always be better than us at this” — has two versions. The strongest version is the asymmetry-everywhere claim: that frontier dominates a custom stack on every corpus. This version is the one the experimental work bears on most directly. Microsoft Copilot for Microsoft 365, Azure OpenAI with Azure AI Search, and the local configurations tested here are the same architecture: a language model with retrieval over a corpus and grounding prompts. The difference is which model sits inside it. Super 120B-A12B with custom RAG, running on A\$10k Spark hardware, lands at seven out of eight competitive, and Lightning 30B-A3B at eight out of eight, against Claude Opus 4.7 and GPT-5 on the test corpus, with correct deceptive-OOC handling. The premise that a custom stack cannot compete is not an argument about architecture, since the architecture is common. It is an argument about the model, which makes it testable.

The deployment-relevant version is different, and is where most of my colleague's intuition lives: “Microsoft Copilot grounded on our actual SharePoint will outperform anything we build, on our actual institutional content.” This version is the case where frontier models are strongest — handling messy multi-author cross-document context that pre-training has rich exposure to — and where custom RAG over a curated subset is most fragile, because the curation step that makes the test corpus work doesn't scale to “all of SharePoint.” My experimental

work does not test this version. What the experimental work establishes is that the strongest version of the asymmetry claim is empirically contested; the deployment-relevant version remains an open empirical question, and my colleague is right to raise it.

The data-locality part — “they have all our email and everything is on SharePoint and Azure anyway” — is a governance question, not a capability question. Auditability against a trusted corpus, which is what this report's institutional argument turns on, is a property of the architectural commitment (Version 1 with grounded prompts). Microsoft Copilot's defaults are tuned for general productivity, not for high-stakes auditability; whether Copilot grounded on SharePoint produces trustworthy answers on regulator-facing questions is empirically unverified. Anyone deploying Copilot on high-stakes content is making the same architectural bet as a custom RAG stack, with less control over prompt engineering and retrieval discipline.

The make-versus-buy part — “why would we build our own system?” — is a strategic-economic question that depends on application class. For light productivity uses (drafting, internal summarisation, meeting notes), Microsoft Copilot is plausibly the right answer for many institutions. For high-stakes uses where output ends up in regulator filings or sector-defining analysis, the audit-trail, prompt-engineering control, and architectural-discipline arguments push the other way — and those are not capability arguments. The report's argument is about distinguishing these application classes.

What this small-scale attempt does and does not establish

The document's central claim is that a senior domain expert with appropriate hardware and methodology can build deployable AI capability on a curated domain corpus, that this capability is competitive with frontier-API alternatives at the same hardware tier on tasks within the corpus's domain, and that the architecture has the failure-mode handling required for high-stakes use.

Two weeks of distributed attention, on A\$10k of hardware, with a single-author technical corpus and substantial chatbot AI assistance, has established the existence claim. There is a configuration of model, retrieval, and prompt structure that produces deployment-quality output on a domain-specific testbed, including correct refusal under engineered deceptive-retrieval conditions, competitive with frontier-API alternatives at the same hardware tier. Three implementation points were tested at that tier, and two of the three cleared the deployment-credibility threshold the design pre-committed to. The methodology required is more demanding than the surface narrative of the public literature suggests: the first attempt failed in a structurally informative way, and the architecture that worked was not the one a reader following the public literature would have tried first.

One asymmetry between the controlled comparison and the deployment claim should be explicit. The experiment held retrieval and prompt structure constant across configurations, which is sound experimental hygiene and makes the result a strict capability comparison. The deployment claim this document's argument rests on is different: practitioner-curated retrieval and prompts on the local side, vendor-curated everything on the frontier side. The local stack has more degrees of freedom in the deployment comparison than the controlled experiment allowed it. That makes the controlled comparison conservative rather than generous, because holding retrieval and prompt structure constant strips the local stack of the curation advantage that is part of why the institutional argument prefers it. A reader importing the controlled comparison as the deployment comparison reads the experimental result as weaker than it is.

What the work has not established is the generalisation claim. The RAG test was on a textbook in a domain where frontier models have substantial pre-training presence; on multi-author multi-organisation corpora where that presence is much weaker, the result may differ. The deployment-deception probe was a single engineered question over six samples, and in production, deceptive-retrieval conditions are the normal state rather than an engineered exception; whether the architecture's refusal discipline holds at that frequency is a hypothesis requiring testing. Version 1 was the only architectural commitment tested. And the practitioner-variance dimension — what a different practitioner with a different competence configuration would experience — is unaddressed by a sample of one.

Interaction for demonstration purposes runs through AnythingLLM, an open-source interface that discovers models from Ollama and builds its own retrieval index over an uploaded document set. Responses carry the documents drawn on and the number of passages from each, alongside the model, elapsed time and throughput. How the interface is configured shapes what it does: a query mode that restricts responses to the uploaded documents, a limit on how many passages reach the model, and a relevance floor below which a passage is not returned. Those are context-engineering decisions (Section 1.4) made in a settings panel rather than in code. Everything reported in this part was run through scripts, not through that interface, so nothing about the demonstration setup is evidence for the claims above. It appears here because the question a reader is most likely to have at this point is what using such a system looks like, and the answer is a browser tab with a source panel.

The deployment claim ultimately rests on a different experiment: the same architecture over multi-author, multi-organisation corpora — RACE reports, AEMO documents, Australian energy-sector publications — where vocabulary-match-without-substance is the normal condition rather than an engineered probe. The textbook experiment establishes that the architecture has the right failure-mode handling under disciplined conditions on a clean corpus. Whether that handling survives the corpus type the argument addresses is untested. A custom benchmark of roughly thirty open-ended questions, designed to detect substantive capability gain rather than surface mimicry, would be the right instrument for quantitative claims. Until that evidence exists, the deployment claim stays conditional.

The reading the experimental work supports is that this report's central claim has not been invalidated under conditions designed to invalidate it. That is weaker than saying the claim is validated. Two weeks of small-scale attempt (plus an additional 90 minutes four months later) with disciplined methodology can show that the claim is plausible and that the architecture has the properties the institutional argument requires. It cannot show that the claim holds at scale, across corpora, across practitioners, or across application classes.

What the work changes is what kind of argument the report is making. Before it, the central claim was a plausibility argument from public evidence: this capability is now within reach for a domain expert with appropriate hardware and methodology. After it, the claim is plausibility plus existence — within reach because it has been reached, at small scale, by one practitioner working under disciplined conditions, with the failure modes the institutional architecture treats as structural turning out to be the ones that emerged.

If this report is right that practitioner capability is a missing fifth position in the public conversation about AI infrastructure, then accounts of practitioners trying to inhabit that position — including the failures, the architectural pivots, the dependency on AI assistance for the layers the practitioner does not have, and the methodological discipline that makes the work interpretable — are part of how that conversation gets started.

- Allen Institute for AI (AI2) (2025–2026). OLMo 3 model release, November 2025; OLMo 3.1 update, December 2025; Olmo Hybrid, March 2026. Retrieved from <https://allenai.org/olmo>
- Allen Institute for AI. (2025). *NSF and NVIDIA award Aiz a combined \$152M to support building a national level fully open AI ecosystem*. 14 August 2025. <https://allenai.org/blog/nsf-nvidia>
- Artificial Analysis (2026). *Artificial Analysis Intelligence Index*. Independent evaluation across nine benchmarks. Retrieved from <https://artificialanalysis.ai>
- Australian Energy Market Commission / Energy Security Board (2025). *National Consumer Energy Resources Roadmap*.
- Australian Energy Market Commission (2026). *Advice to the Energy and Climate Change Ministerial Council on regulatory pathways for a data centre energy obligation*. Delivered 28 July 2026, publicly released 5 August 2026.
- Australian Energy Market Operator (2026). *2026 Electricity Statement of Opportunities*. August 2026.
- Australian Government (2025). *National AI Plan*. Department of Industry, Science and Resources, December 2025. Retrieved from <https://www.industry.gov.au/NationalAIPlan>
- Australian Government, Department of Industry, Science and Resources (2026). “AI Accelerator initiative kicks off with funding for industry-led research.” Announcement regarding CRC-P round 19 (applications closing 12 May 2026) and CRC round 28 (opening 2027), 18 March 2026.
- Australian Government, Department of Industry, Science and Resources (2026). *Expectations of Data Centres and AI Infrastructure Developers*. 23 March 2026. Retrieved from <https://www.industry.gov.au/NationalDataCentreExpectations>
- Australian Government and Anthropic (2026). *Memorandum of Understanding between the Australian Government and Anthropic on Collaboration on AI Opportunities*. Signed 1 April 2026; the first MoU under the National AI Plan. Retrieved from <https://www.industry.gov.au/publications/memorandum-understanding-between-australian-government-and-anthropic-collaboration-ai-opportunities>
- Australian Institute of Company Directors (AICD) and Governance Institute of Australia (2024–2025). *A Director’s Introduction to AI* and associated governance guidance, including joint materials on privilege compromise in AI-assisted board processes.
- Becker, J., Rush, N., Barnes, E., & Rein, D. (2025). *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*. arXiv:2507.09089. <https://metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/>
- Christou, D. and Tsoumakas, G. (2026). “Sub-Billion, Super-Frontier: Small Language Models Rival Zero-Shot Frontier LLMs on General and Literary Relation Extraction.” arXiv:2606.22606, 21 June 2026.
- Commonwealth Scientific and Industrial Research Organisation (CSIRO) (2024). *Artificial Intelligence Foundation Models: Industry Enablement, Productivity Growth, Policy Levers and Sovereign Capability Considerations for Australia*. March 2024. Retrieved from <https://www.csiro.au/en/research/technology-space/ai/ai-foundation-models-report>
- Dunsford, G. (2026). “Some innovations don’t just improve the way we work — they change what’s possible. And in energy, that matters.” LinkedIn post, April 2026. Public statement by AEMO’s Group CIO/CTO describing AEMO’s deployment of Microsoft AI capabilities at Microsoft AI World Tour Sydney.
- Emberson, L. (2025). “Open-weight models lag state-of-the-art by around 3 months on average.” Epoch AI, 30 October 2025. Retrieved from <https://epoch.ai/data-insights/open-weights-vs-closed-weights-models>
- Engineers Australia (2025). *Artificial Intelligence and the Engineering Profession*. January 2025.
- Epoch AI (2025). “Frontier AI capabilities can be run at home within a year or less.” Epoch AI Data Insight, 15 August 2025. Retrieved from <https://epoch.ai/data-insights/consumer-gpu-model-gap>
- Goldenfein, J., Parker, C., and Weatherall, K. (2025). “Australia’s national AI plan has just been released. Who exactly will benefit?” *The Conversation*, 2 December 2025.
- Kotha, S., Springer, J. M., and Raghunathan, A. (2024). “Understanding Catastrophic Forgetting in Language Models via Implicit Inference.” In *The Twelfth International Conference on Learning Representations (ICLR 2024)*, Vienna. arXiv:2309.10105. Retrieved from <https://openreview.net/forum?id=VrHiF2hsrm>
- Law Council of Australia, Law Society of New South Wales, Law Institute of Victoria, and Law Society of Western Australia (2024–2025). *Joint Statement on the Use of Generative AI in Legal Practice*.
- Merchant, B. (2023). *Blood in the Machine: The Origins of the Rebellion Against Big Tech*. New York: Little, Brown and Company.

Merrill, W., Li, S., Romero, J., Svete, A., Costello, C., et al. (2026). “Olmo Hybrid: From Theory to Practice.” arXiv:2604.03444.

Nagle, F. and Yue, D. (2025). “The Latent Role of Open Models in the AI Economy.” SSRN Working Paper 5767103, 18 November 2025. Retrieved from <https://ssrn.com/abstract=5767103>. Summarised in MIT Sloan Management, “AI open models have benefits. So why aren’t they more widely used?” (February 2026).

NSW Government (2026). *Electricity Infrastructure Investment Amendment Bill 2026*. Introduced to the Legislative Assembly 5 August 2026.

NSW Government (2026). *NSW Data Centre Policy Framework*. Infrastructure NSW, August 2026. Consultation closing 14 September 2026.

NVIDIA (2025–2026). DGX Spark and DGX Station product documentation. Retrieved from <https://www.nvidia.com/en-au/products/workstations/dgx-spark/> and <https://www.nvidia.com/en-au/data-center/dgx-station/>

NVIDIA (2025–2026). Nemotron 3 family documentation and model cards: Nano (December 2025), Super (March 2026), Ultra (June 2026), Nemotron 3.5 Lightning (August 2026); Nemotron Coalition announcement, March 2026. Technical reports arXiv:2512.20848 (Nano) and arXiv:2606.15007 (Ultra).

Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J. E., Kadous, M. W., & Stoica, I. (2025). RouteLLM: Learning to Route LLMs from Preference Data. *The Thirteenth International Conference on Learning Representations (ICLR 2025)*, Singapore. <https://openreview.net/forum?id=8sSqNntaMr>

OpenRouter (2026). Model pricing and hosting comparison. Retrieved from <https://openrouter.ai>

Paradis, E., Grey, K., Madison, Q., Nam, D., Macvean, A., Meimand, V., Zhang, N., Ferrari-Church, B., & Chandra, S. (2024). *How Much Does AI Impact Development Speed? An Enterprise-Based Randomized Controlled Trial*. IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP 2025), 618–629.

Peng, S., Kalliamvakou, E., Cihon, P., and Demirer, M. (2023). “The Impact of AI on Developer Productivity: Evidence from GitHub Copilot.” arXiv:2302.06590.

Prehn, J., Carlson, B., Walter, M., Lovett, R., and Maher, B. (2026). “Indigenous Data Sovereignty and the Governance of Artificial Intelligence: Toward Equitable Benefit for Indigenous Peoples.” *Journal of Sociology*. DOI: 10.1177/14407833261460137.

Royal Australian College of General Practitioners (RACGP). *Position Statement on Artificial Intelligence in Primary Care*.

Sontag, E.D. (2026). “Mathematics in the Age of AI: From Theorem Proving to Theorem Mining.” Substack, 27 March 2026. Retrieved from <https://eduardosontag.substack.com/p/mathematics-in-the-age-of-ai-from>

Sovereign Australia AI (2025–). Project documentation and model releases. Retrieved from <https://sovereign-au.ai>

Stanford University Human-Centered AI Institute (2025). *AI Index Report 2025*. Stanford, CA: Stanford HAI.

Strategic Examination of R&D independent expert panel (2025). *Ambitious Australia: Strategic Examination of R&D Final Report*, December 2025. Publicly released 17 March 2026.

Xue, C., Wang, Y., Liu, M., Liang, D., Han, X., Liu, P., Wu, X., Lu, C., Jiang, L., Lu, Y., Shi, H., Liang, S., Peng, M., and Salim, F.D. (2026). “Why Supervised Fine-Tuning Fails to Learn: A Systematic Study of Incomplete Learning in Large Language Models.” *In Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30186–30213. Association for Computational Linguistics. Preprint: arXiv:2604.10079.

A note on authorship and method. This report was drafted over several months using the workflow described in Sections 1.5 and 1.6: AI assistance (Anthropic Claude) for drafting, structural challenge, source retrieval, and iterative refinement, with the author directing the work, evaluating each output, and making every strategic, factual, and framing judgement. The author’s typed input to the AI over the drafting period substantially exceeds the word count of the final document — most of it critique, correction, pushback, and redirection. The analytical framework, the domain knowledge, the strategic positioning, and the arguments are the author’s. The execution of the drafting was compressed by the tool. See disclosure on page 1 regarding the author’s affiliations and interests.