TECHNICAL ANALYSIS · 22 min read

Small, task-specific models

Specialised models, distillation, routing and right-sizing limits.

Document language : English

Document type : Technical synthesis and reference architecture · V2

Targeted reading review : 2026-10-15

Reading notice — 15 October 2026

> *Document type and scope: a synthesis of sources and a proposed reference architecture. This is not a peer-reviewed publication, a report of a LinkTec experiment or a client reference.*
>
> *Scope correction: manuscript wording such as "deploys", "builds", "audit-ready", "sovereign" and similar terms describes architecture objectives requiring project-specific verification. It attests no client deployment, operated equipment, certification or LinkTec result. Working hypotheses are neither completed research nor a commitment to publish.*
>
> *Quoted quantitative results belong to their authors and evaluation settings. General claims, recommendations and thresholds are not guarantees for every workload. The reading review and checks of principal references are not experimental replication or exhaustive scientific validation.*
>
> *Execution is on client premises or a cloud selected with the client. A model's origin, size and openness determine neither execution location nor commercial rights. Versions, licences, access and dependencies require verification. No US-based model API without the client's request.*
>
> *Clarifications: "small" guarantees neither quality, licensing status nor sovereignty. Phi-4, Qwen, SmolLM2, routing and distillation comparisons depend on their authors' versions, tasks, hardware and evaluation settings. This note guarantees no cost ratio, benchmark score or memory threshold. Market forecasts are not observed results. Configuration, RAG and fine-tuning are selected according to needs and tests; fine-tuning and external API use are not automatic. Model and distillation-data rights still require verification.*

The right model is the smallest that solves the task.

LinkTec Labs · Research Note N°04 · Applied AI & Data · Edition 2026

Abstract

The default reflex of routing every request to the largest available frontier model is, increasingly, an anti-pattern of cost, latency, energy and sovereignty. A quieter shift is underway: sub-10-billion-parameter models now match 2024-era frontier systems on most benchmarks, distilled reasoning models let a 14-billion-parameter student outscore proprietary systems on competition mathematics, and a small model fine-tuned for a narrow task routinely beats a much larger general one on that task. For the agentic workloads now reaching production, which consist mostly of repetitive, specialised sub-tasks, this changes the economics entirely. This note sets out the case for right-sizing: what "small" means in 2026, why agents make small models inevitable, how specialisation (distillation, fine-tuning, quantisation, and now reasoning transfer) makes them excellent, how multi-adapter serving makes a whole portfolio of them cheap to operate, and how routing sends each query to the cheapest capable model, often beating the single best model while cutting cost by large multiples. We are equally explicit about where small models fail, because right-sizing without honest limits is just under-provisioning. We close with the model-portfolio reference architecture proposed by LinkTec Labs, its data flywheel, and five working hypotheses that define our research agenda.

---

01 · The "biggest model" reflex: why default-to-frontier is an anti-pattern

The most expensive habit in applied AI is sending every request to the largest model available. It feels safe: the frontier model can do anything, so why think harder? But that reflex pays a frontier price for tasks a far smaller model would solve identically, on every call, forever. At the scale of an agentic system that issues thousands of model calls per workflow, the difference is not a rounding error. It is the difference between a deployment that pays for itself and one that never does.

The counter-position is now backed by serious research. NVIDIA's 2025 position paper argues, plainly, that small language models are the future of agentic AI: sufficiently capable for the bulk of agent calls, inherently better suited to specialised repetitive work, and necessarily more economical [1]. The economic gap is not marginal. Serving a 7-billion-parameter model is, by their accounting, roughly ten to thirty times cheaper in latency, energy and compute than serving a 70-to-175-billion-parameter one [1]. The market has drawn the same conclusion: Gartner predicts that by 2027, organisations will use small, task-specific models at least three times more than general-purpose LLMs, citing precisely the accuracy and cost pressures this note describes [10]. When the same task can be done at a tenth of the cost with no measurable loss of quality, doing it at full price is not caution. It is waste.

The thesis of this note is a single engineering principle, and it is deliberately unglamorous:

> ***The right model is not the most capable one available. It is the smallest one that solves the task to the required standard, measured.***

Everything that follows (the 2026 capability landscape, specialisation, routing, the economics, the limits) is in service of making that principle operational, and of showing that it is now not only cheaper but frequently more accurate than the default.

Key terms

A shared vocabulary for the rest of this note. Specialists may skip ahead.

Term	Meaning
SLM	Small Language Model: one that runs on common consumer hardware and responds fast enough for a single user (often under ~10B parameters).
Parameters	The learned weights of a model; a rough, imperfect proxy for size, cost and capability.
Distillation	Training a small "student" model to imitate a large "teacher", transferring capability into a fraction of the size.
Reasoning transfer	Distilling not answers but chains of reasoning, so the student inherits the teacher's problem-solving behaviour.
Fine-tuning / LoRA	Cheaply adapting a base model to a specific task; LoRA trains small adapter weights instead of the whole model.
Quantisation	Storing weights at lower precision (e.g. 4-bit) to shrink memory and run on commodity or edge hardware.
Multi-adapter serving	Hosting one base model and swapping many LoRA adapters on it, so a whole portfolio shares one deployment.
Model routing	Directing each query to the cheapest model capable of answering it correctly.
Cascade	Trying a cheap model first and escalating to a stronger one only if a quality check fails.
Thinking budget	The inference-time reasoning tokens a model is allowed; a second capacity dial alongside parameter count.
On-device / edge	Running the model locally (laptop, phone, on-prem box) rather than in a remote cloud.
Right-sizing	Matching model capacity to task difficulty, rather than over-provisioning by default.

02 · What "small" means now: the capability inversion

The word "small" misleads, because capability has stopped tracking size. A small model in 2026 is defined functionally, not by a parameter count alone: one that runs on common consumer hardware and answers fast enough for a single user [1]. What has changed is what such models can do. Sub-10-billion-parameter models now match or exceed 2024-era frontier systems on most benchmarks, for well-understood reasons: heavy training on curated synthetic data, distillation from frontier teachers, and architectural refinement.

The examples are concrete. Microsoft's Phi-4, at 14 billion parameters, reaches roughly 84.8% on MMLU and surpasses much larger models on mathematical reasoning [5]. Alibaba's Qwen3 family, down to a few billion parameters, ships strong native tool-calling, with a 4-billion-parameter variant scoring in the low 80s on MMLU-Redux [6]. Hugging Face's SmolLM2, at 1.7 billion parameters, runs in as little as 6 GB of memory [7]. On the deployment side, flagship phones and AI PCs now ship neural accelerators sized exactly for this class, and Apple's on-device foundation model is itself a roughly 3-billion-parameter SLM: the hardware market has already voted.

The capability inversion has a second axis that did not exist two years ago: inference-time reasoning. Scaling the compute a model spends thinking at answer time can substitute for parameters, and in a growing range of settings a small model given more thinking tokens outperforms a much larger model answering directly [11]. Distillation transfers this too. DeepSeek's R1 distillations are the landmark result: a 14-billion-parameter student trained on the reasoning traces of a large teacher outscores OpenAI's o1-mini on competition mathematics (AIME 69.7%, MATH-500 93.9%), setting state-of-the-art results for dense models of that size [12]. Parameter count is no longer a proxy for capability; the question "is this model big enough?" has been replaced by "is this model, with this thinking budget, right for this task?"

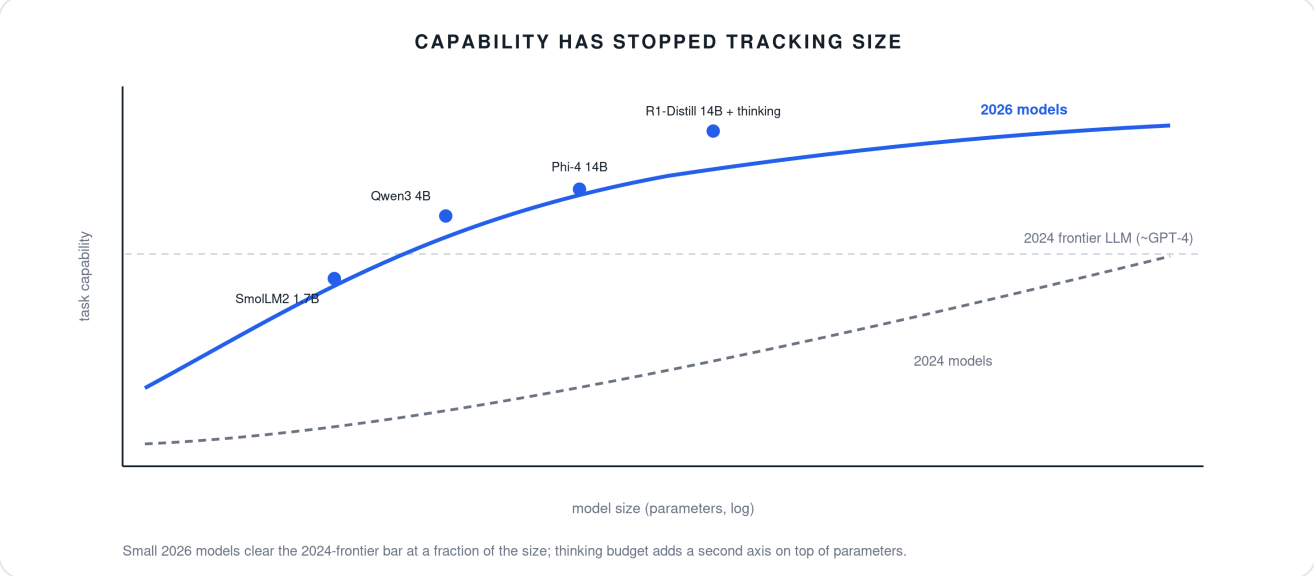


Fig. 1 — The capability inversion. By 2026, sub-10B models clear the 2024-frontier bar through synthetic-data training, distillation and architecture, and reasoning transfer adds a second axis: thinking budget can substitute for parameters [5][6][7][11][12]. Schematic.

03 · Why agents make small models inevitable: specialised, repetitive, narrow

The case for small models is strongest exactly where AI is heading: agents. An agentic workflow is not one grand act of reasoning; it is a long sequence of small, specialised, low-variation sub-tasks: extract a field, classify an intent, format a response, decide a tool call, validate an output. Each of these is narrow and repetitive, and narrow repetitive work is precisely what a small specialised model does best, and what a frontier model is absurdly over-qualified for [1].

This motivates the heterogeneous agent: a system in which a small model handles the overwhelming majority of calls by default, and a large model is invoked only for the rare step that needs open-ended reasoning. The frontier model becomes a specialist consultant called in for hard cases, not the general practitioner seeing every patient. For the organisation, this is the move that makes agents affordable at production volume, because it stops paying frontier rates for clerical work.

> ***The reframing.*** *An agent is not one model doing everything. It is a small team of mostly-small models, each doing one thing well, with a frontier model on call for the hard step. Capacity is allocated to difficulty, not sprayed uniformly [1].*

---

04 · How small models get good: specialisation, not scale

A small model is not simply a worse large one. Three well-established techniques, plus one piece of serving infrastructure, turn a compact base model into a portfolio that beats a far larger generalist on its own ground.

Distillation trains a small "student" to imitate a large "teacher", compressing capability into a fraction of the parameters; it is a foundation of nearly every strong small model today [8], and, as the R1 results show, it now transfers reasoning behaviour, not just answers [12]. **Fine-tuning**, and especially LoRA, cheaply adapts a base model to a narrow task by training small adapter weights rather than the whole network, making per-task specialisation affordable [9]. **Quantisation** stores weights at lower precision, commonly 4-bit, shrinking memory enough to run the model on commodity or edge hardware with little quality loss.

The piece most treatments omit is what makes the portfolio cheap to operate rather than merely cheap to train. Serving fifty specialised models does not mean running fifty deployments: multi-adapter serving systems such as S-LoRA hold one base model on the GPU and page thousands of LoRA adapters over it on demand, with throughput several times higher than naive adapter serving [13]. The marginal cost of one more specialist collapses toward the cost of storing its adapter, a few tens of megabytes. That single fact converts "a model per task" from an operations nightmare into a routine pattern.

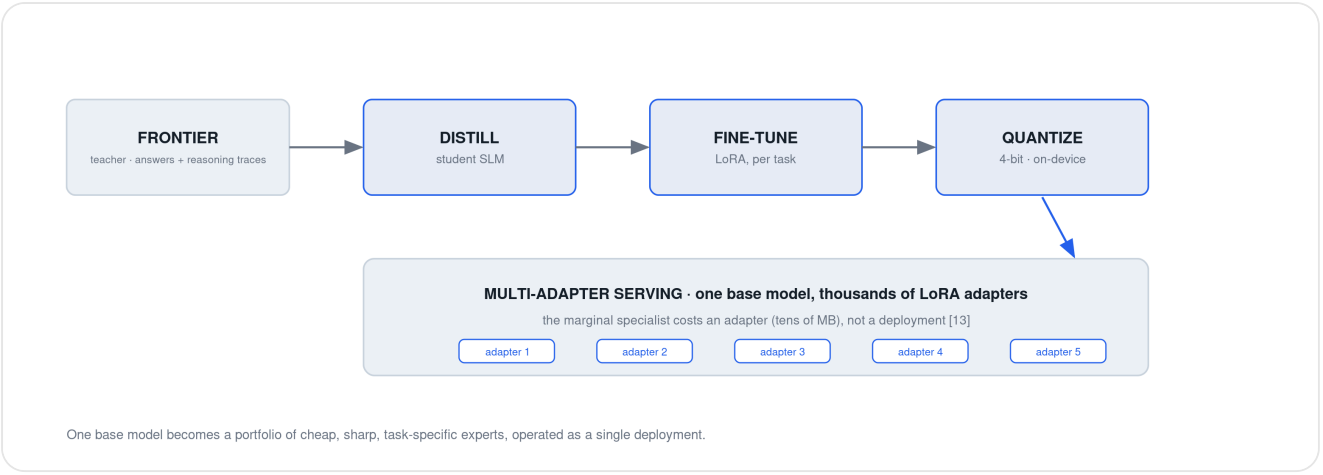


Fig. 2 — The specialisation pipeline. Distillation [8][12] moves capability (and reasoning) into a small student; LoRA fine-tuning [9] sharpens it per task; quantisation fits it on commodity or edge hardware; multi-adapter serving [13] runs the whole portfolio on one deployment.

05 · Routing: the right model per query

If you have a portfolio of models, the remaining question is which one to use for a given request. This is model routing: a lightweight router classifies each query and sends it to the cheapest model capable of answering it correctly, escalating only when needed. Two families have matured. Predictive routers, exemplified by RouteLLM, decide before inference using a learned estimate of difficulty [2]. Non-predictive cascades, exemplified by FrugalGPT, try a cheap model first and escalate only if a quality check fails [4].

The measured results are large. RouteLLM reports cost reductions of over 85% on MT-Bench while retaining 95% of GPT-4's quality, by sending only the hard queries to the expensive model [3]. FrugalGPT reports up to 98% cost reduction at equal or better accuracy on specific benchmarks [4]. In production, most enterprise teams report a more sober but still decisive 40 to 70% saving. And there is a subtler prize: a well-designed router can beat even the single most capable model, because it can lean on each model's particular strengths rather than forcing one model to be best at everything.

Reasoning models add a third routing dimension. The choice is no longer only which model but how long it may think: many requests are served correctly by a small model answering directly, some need the same small model with an extended thinking budget, and only the residue needs the frontier [11]. Treating the thinking budget as part of the routing decision captures savings that model choice alone leaves on the table.

One warning belongs beside the numbers: a router is a model, and it fails like one. A router that silently mis-routes hard queries to the cheap path is a hidden quality regression distributed across your traffic, which is why routing decisions must be continuously validated against measured outcomes, not trusted after a one-off calibration.

10-30×	>85%	3×
Cheaper to serve a 7B model than a 70-175B one, in latency, energy and compute [1].	Cost reduction on MT-Bench via routing, keeping 95% of frontier quality [3].	More use of task-specific small models than general LLMs by 2027, per Gartner [10].

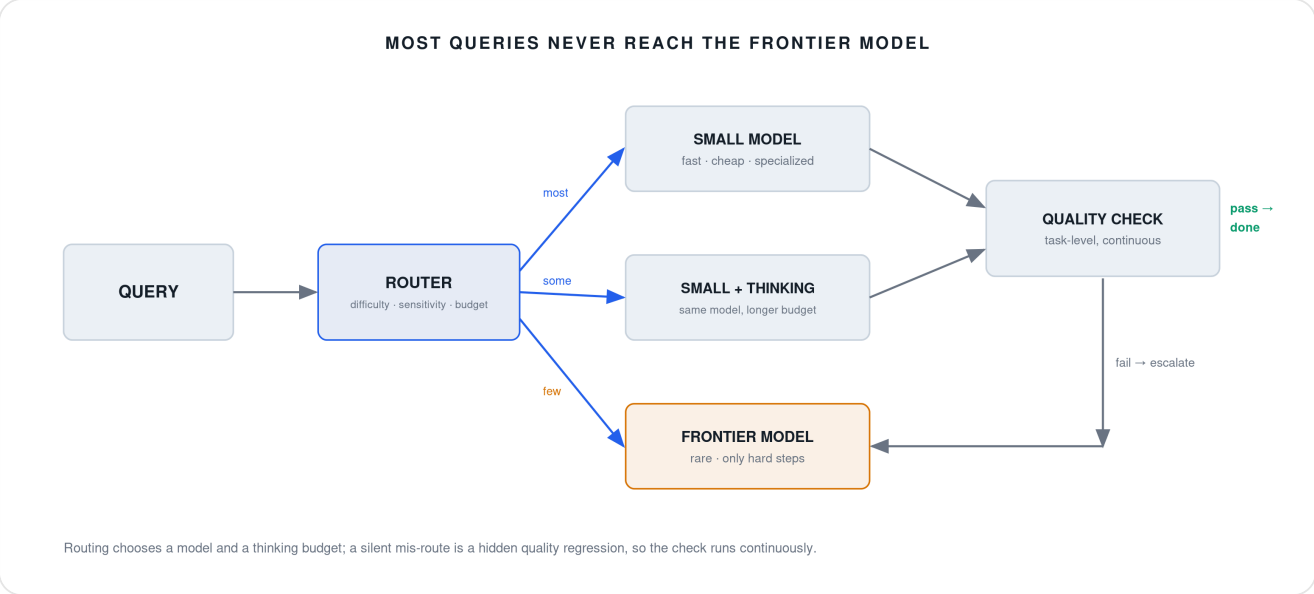


Fig. 3 — Routing and cascade. A router sends most queries to a small, specialised model, may grant the same model a longer thinking budget, and escalates to a frontier model only when a quality check fails, capturing most of the cost saving [2][3][4][11].

06 · Where small models fail: the honest limits of right-sizing

Right-sizing is only credible if its limits are stated as plainly as its wins. Four are established.

Error compounding in long chains. An agent step with 95% reliability compounds to roughly 77% over five dependent steps and 60% over ten. Small models tend to fail exactly here: not on the single narrow call, but through slightly noisier intermediate outputs (a mis-specified tool argument, a drifting summary) whose errors accumulate downstream. The mitigation is architectural: validate at step boundaries, keep sub-tasks independent where possible, and escalate the orchestration-critical steps rather than the average ones.

Long context and open-ended breadth. Deep research tasks, very long documents and genuinely open-ended synthesis remain the territory of larger models; small models degrade earlier as context grows and have thinner long-tail world knowledge. A portfolio without a frontier fallback is not right-sized; it is under-provisioned.

Benchmark inflation. MMLU-class scores are the weakest possible evidence for task fit: public benchmarks leak into training data, and a model tuned to clear a leaderboard bar can still fail your invoice-extraction task. The only score that matters is measured accuracy on your task distribution, which is why the evaluation loop in section 08 is not optional.

Fragmentation overhead. Every specialist added to the portfolio is another artefact to version, evaluate, monitor and re-train as tasks drift. Multi-adapter serving removes the hardware cost [13]; it does not remove the governance cost. A portfolio should grow one measured specialist at a time, not explode by enthusiasm.

None of these reverses the argument. They define its boundary: small by default, frontier by exception, and evaluation deciding which is which.

07 · The economics, and the sovereignty bonus: cheaper, greener, closer to home

Right-sizing pays in more than one currency. The direct saving is cost and latency: an order of magnitude or more per right-sized call [1], compounding across every step of every agent run. The second is energy: the same factor of ten to thirty applies to power draw, which matters both for unit economics and for sustainability reporting, and increasingly for regulators asking about the footprint of AI deployments. The third, and for European organisations often the decisive one, is sovereignty. A small model can run on-device or on-premise, which means the data it processes never has to leave the building. Right-sizing and the sovereign-by-design architecture of Research Note N°03 are the same move seen from two angles: the smaller the model, the easier it is to keep both the data and the computation inside your own walls.

Use	Best choice	Why
High-volume, narrow sub-task	Specialised SLM	10-30× cheaper; can run on-device for sovereignty [1]
Mixed difficulty at scale	Router / cascade	Cheap by default, escalate rarely; 40-85% saving [3][4]
Hard reasoning, bounded domain	Small reasoning model + thinking budget	Reasoning transfer puts competition-grade maths in 14B [11][12]
Rare, open-ended reasoning	Frontier LLM	Worth its cost only where capability is actually needed
Regulated / personal data	On-device SLM	Data never leaves the perimeter (see Note N°03)

08 · A reference architecture: a model-portfolio proposal

The pieces compose into a portfolio managed by a router and kept honest by evaluation. Rather than a single model behind an endpoint, the system holds a registry of small, task-specialised models served as adapters on shared bases, a router that picks a model and a thinking budget per query by difficulty and sensitivity, a frontier fallback for the hard or rare cases, and a continuous evaluation loop that re-checks routing decisions against measured quality and re-specialises models as tasks drift.

The loop has a fourth element that turns cost into capability: the data flywheel [1]. Every call that escalates to the frontier model is logged (query, context, response, outcome) and those traces are exactly the training data for the next specialist. Escalations are clustered, the recurring clusters are distilled into new or updated adapters, and the traffic that once paid frontier rates migrates to the portfolio. A healthy deployment shows a measurable signature: its frontier-escalation rate declines month over month while task accuracy holds or improves. The frontier model stops being a permanent cost centre and becomes the teacher that continuously trains its own replacements.

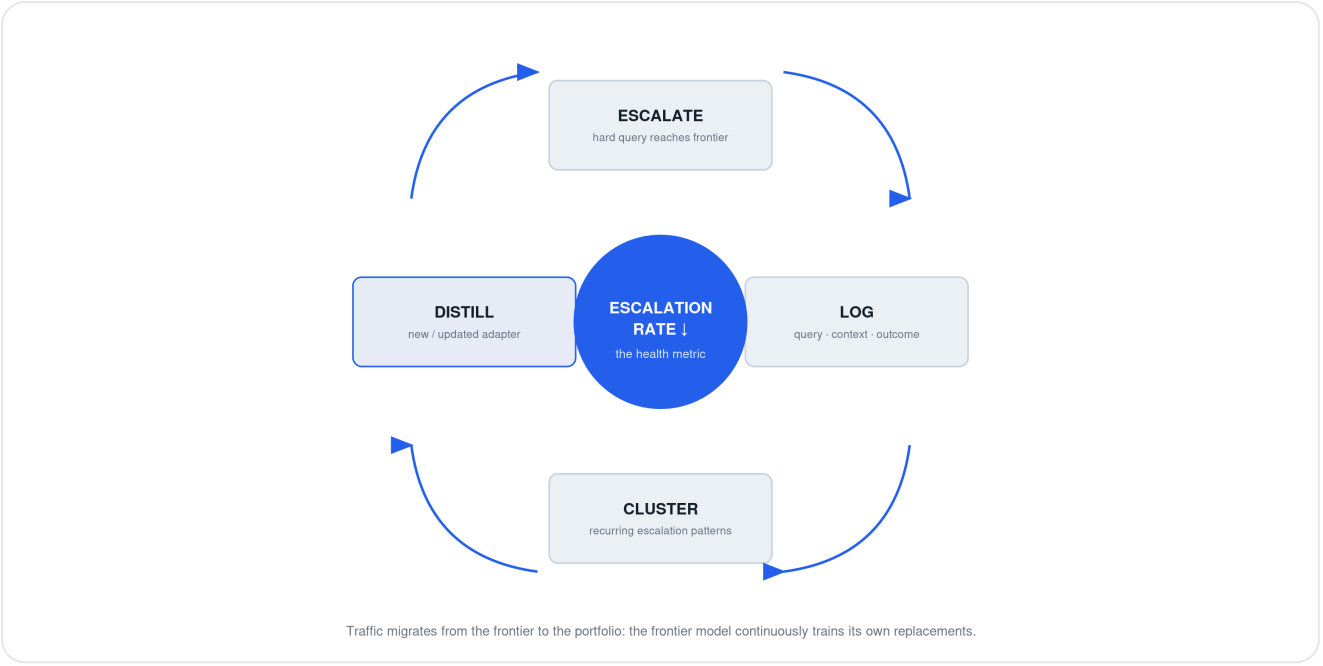


Fig. 4 — The data flywheel. Frontier escalations are logged, clustered and distilled into new specialised adapters; traffic migrates from the frontier to the portfolio, and the escalation rate becomes the health metric of the deployment [1][8][12].

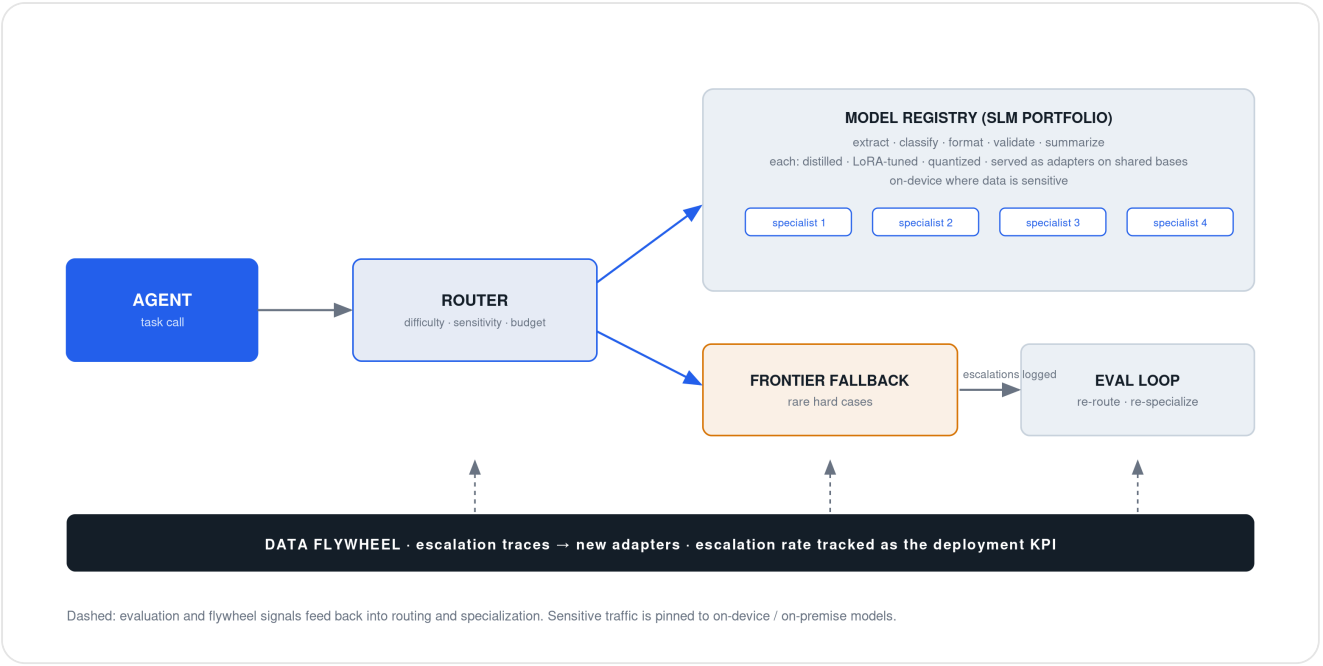


Fig. 5 — LinkTec Labs model-portfolio architecture. A router selects a specialised small model and a thinking budget per task by difficulty and sensitivity, falls back to a frontier model for rare hard cases, serves the portfolio as adapters on shared bases [13], and an evaluation loop with the data flywheel keeps routing and specialisation current.

Five principles govern it. **Capacity follows difficulty**: the default is the smallest capable model, and frontier capability is summoned, not assumed. **Specialisation is the lever**: distillation, LoRA and quantisation turn one base into many sharp experts, served together. **Routing is measured, not guessed**: the router's decisions are validated against quality continuously, because a router that mis-routes silently is worse than no router. **Escalations are training data**: the flywheel converts every frontier call into tomorrow's specialist. **And small is sovereign**: wherever data is sensitive, the chosen model runs on-device or on-premise, closing the loop with Note N°03. The first four buy efficiency that compounds; the fifth buys trust.

09 · Open problems: five working hypotheses

Right-sizing is an engineering practice with open research questions at its edges. These five hypotheses define LinkTec Labs' agenda; each is falsifiable, and no completed experiment or publication schedule is asserted.

H1. The routing objective should be cost per correct answer, not cost per token. Routers today optimise proxy scores calibrated offline. We hypothesise that a router trained on production telemetry against task-level outcomes (was the extraction right, did the workflow complete) dominates benchmark-calibrated routing, and that the gap widens as task distributions drift from public benchmarks.

H2. Thinking budget is a routing dimension, not a model property. We hypothesise that three-way routing (small direct, small with extended reasoning, frontier) strictly dominates two-way model routing at equal quality, because a large share of "hard" queries are hard in a way that more thinking, not more parameters, resolves [11][12].

H3. Escalation rate is the KPI of a healthy portfolio. If the flywheel works, the share of traffic reaching the frontier model should decay toward an irreducible floor of novel cases. We hypothesise this decay curve is measurable, predictable per domain, and the single best leading indicator of deployment economics; a flat curve means the flywheel is broken.

H4. Drift management can be automated per adapter. Specialists go stale as tasks evolve. We hypothesise that per-adapter canary evaluations, triggered by distribution-shift detection on inputs, catch degradation before users do, and that re-specialisation from fresh escalation traces restores accuracy at a small, bounded cost, making a fifty-specialist portfolio governable by a small team.

H5. Sensitivity is a routing input before cost is. For regulated data, the feasible model set is determined by where each model runs (on-device, on-premise, cloud) before any cost-quality trade-off applies. We hypothesise that sovereignty-aware routing (data classification first, economics second) can be enforced mechanically at the router, giving auditors a checkable guarantee that sensitive data never reached an external endpoint (see Note N°03).

10 · Practitioner's playbook: what to do on Monday

Stop defaulting to frontier. Make "smallest capable model" the default and force a reason to escalate [1].

Profile your agent's calls. Most are narrow and repetitive; those are SLM targets, not frontier ones [1].

Specialise, don't scale. Distil, LoRA-tune and quantise a base model per task [8][9]; for reasoning-heavy niches, distil reasoning traces, not just answers [12].

Serve the portfolio as adapters. One base, many LoRAs, one deployment; the marginal specialist is nearly free [13].

Add a router. Predictive routing or a cascade with a quality threshold captures most of the saving [2] [4]; include thinking budget in the decision [11].

Measure routing, always. A silent mis-route is a hidden quality regression; validate continuously against task outcomes [3].

Log every escalation. Frontier calls are your next training set; cluster them and distil the recurring ones [1].

Respect the limits. Validate at step boundaries in long chains, keep a frontier fallback, and never trust a leaderboard score over your own eval.

Run small models on-device for sensitive data. Right-sizing is also a sovereignty win (Note N°03).

Track cost and energy per task, not just per token. That is where right-sizing shows up.

> ***Bigger is not safer; it is just more expensive by default. The discipline, and the advantage, is using the smallest model that does the job, and proving that it does.***

LinkTec Labs designs right-sized model portfolios (specialised small models, measured routing, sovereign on-device deployment) for organisations that want capable AI without frontier bills. Fourth in our 2026 research series, and the efficiency complement to context (N°01), memory (N°02) and sovereignty (N°03). **linktec.fr / labs**

References

Belcak et al. (NVIDIA Research). *Small Language Models are the Future of Agentic AI.* 2025. arXiv:2506.02153. https://arxiv.org/abs/2506.02153

Ong et al. (LMSYS / UC Berkeley). *RouteLLM: Learning to Route LLMs with Preference Data.* 2024. arXiv:2406.18665. https://arxiv.org/abs/2406.18665

LMSYS Org. *RouteLLM: An Open-Source Framework for Cost-Effective LLM Routing* (cost-reduction figures: >85% MT-Bench at 95% of GPT-4 quality). 2024. https://lmsys.org/blog/2024-07-01-routellm/

Chen, Zaharia & Zou. *FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance.* 2023. arXiv:2305.05176. https://arxiv.org/abs/2305.05176

Abdin et al. (Microsoft Research). *Phi-4 Technical Report.* 2024. arXiv:2412.08905. https://arxiv.org/abs/2412.08905

Qwen Team (Alibaba). *Qwen3 Technical Report.* 2025. arXiv:2505.09388. https://arxiv.org/abs/2505.09388

Allal et al. (Hugging Face). *SmolLM2: When Smol Goes Big.* 2025. arXiv:2502.02737. https://arxiv.org/abs/2502.02737

Hinton, Vinyals & Dean. *Distilling the Knowledge in a Neural Network.* 2015. arXiv:1503.02531. https://arxiv.org/abs/1503.02531

Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models.* ICLR 2022. arXiv:2106.09685. https://arxiv.org/abs/2106.09685

Gartner. *Gartner Predicts by 2027, Organizations Will Use Small, Task-Specific AI Models Three Times More Than General-Purpose Large Language Models.* Press release, April 2025. https://www.gartner.com/en/newsroom/press-releases/2025-04-09-gartner-predicts-by-2027-organizations-will-use-small-task-specific-ai-models-three-times-more-than-general-purpose-large-language-models

Snell et al. *Scaling LLM Test-Time Compute Optimally Can Be More Effective than Scaling Model Parameters.* 2024. arXiv:2408.03314. https://arxiv.org/abs/2408.03314

DeepSeek-AI. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning* (incl. distilled 1.5B-70B checkpoints). 2025. arXiv:2501.12948. https://arxiv.org/abs/2501.12948

Sheng et al. *S-LoRA: Serving Thousands of Concurrent LoRA Adapters.* MLSys 2024. arXiv:2311.03285. https://arxiv.org/abs/2311.03285

© 2026 LinkTec, LinkTec Labs. Research Note N°04. Figures are original schematics by LinkTec Labs, after the cited studies. Quantitative results are attributed to their primary sources; vendor and secondary figures are indicated as such in-text. This note is informational and does not constitute a benchmark claim by LinkTec.

Principal sources consulted

Small Language Models — Belcak et al.

RouteLLM — Ong et al.

Phi-4 — Abdin et al.

Qwen3 — Qwen Team

SmolLM2 — Allal et al.

DeepSeek-R1 — DeepSeek-AI

S-LoRA — Sheng et al.