QUANTUM RESERVE PRESS · TECHNICAL FRAMEWORK SERIES

An Integrated Framework for Comprehensive AI System Development

Architectural Paradigms, Capability Asymmetry, and the Computational Substrate

Dr. Gregory S. Carmichael, DBA
Chief Executive Officer, Quantum Reserve Capital
Lieutenant Colonel, United States Air Force (Retired)

First Edition · July 2026

PhD rigor. Dinner-table clarity. Unambiguous consequence.

Abstract

Contemporary artificial intelligence systems present a puzzle that is more than a curiosity: the same model that produces a competent graduate-level treatment of quantum chromodynamics may fail to count the letters in a short word. Capability is not monotone in human-perceived difficulty. This document treats that asymmetry as the central design constraint for anyone building on these systems rather than as an amusing defect, and it develops three things in sequence. First, a taxonomy of the architectural paradigms currently being combined in pursuit of broader, more reliable machine intelligence—sparse and state-space scaling, structured and reinforcement-trained reasoning, tool-augmented modularity, retrieval grounding, multi-agent composition, and neurosymbolic verification. Second, a five-cause account of why capability asymmetry arises: sub-token opacity introduced by the tokenizer, training distribution mismatch, the compression–retrieval tradeoff inherent to parametric memory, bounded serial depth in a fixed-depth forward pass, and the anthropocentric calibration error that makes us surprised by the wrong failures. Third, an integrated five-layer reference architecture built around two organizing commitments—*capability-aware routing* and a *verification gradient*—together with implementation guidance, evaluation criteria, security and post-quantum considerations, and an analysis of the hardware substrate on which all of it runs. A boundary condition runs through the whole: retrieval grounds a claim in what has been written rather than in what is true, so a widely propagated error arrives with impeccable provenance and passes every provenance-based check. The framework answers this by grading evidence at corpus ingestion and by measuring evidentiary support separately from citation traceability. It is written for regulated, critical-infrastructure, and federal contexts, where the governing question is not how impressive a system is on a good day but how it fails, how that failure is detected, and what it costs when the detection is late.

Keywords: large language models; mixture of experts; state-space models; chain-of-thought; retrieval-augmented generation; neurosymbolic systems; capability asymmetry; tokenization; AI accelerators; quantum computing; post-quantum cryptography; critical infrastructure.

Citation. Carmichael, G. S. (2026). *An Integrated Framework for Comprehensive AI System Development: Architectural Paradigms, Capability Asymmetry, and the Computational Substrate.* First Edition. Quantum Reserve Press.

Document class. Executive technical framework. This is a synthesis and design document, not a report of new empirical results, and not a conformity assessment against any standard. Its central empirical claim (Equation (4)) is stated as a testable conjecture rather than a finding; see Section 8.

Sources and currency. Claims about third-party systems, hardware, and standards reflect public reporting current as of July 2026, are labelled in the References by source status, and should be re-verified before use in a decision. Figures marked *schematic* carry no measured values.

Figures and tables. All figures and tables are original work of the author, produced in TikZ and pgfplots.

Funding and conflicts. Prepared without external funding. The author is Chief Executive Officer of Quantum Reserve Capital and holds interests in ventures that deploy AI infrastructure; no vendor reviewed, funded, or approved this document, and no vendor relationship influenced the selection or assessment of the technologies discussed.

Accessibility. Text layer is extractable and the document is readable in grayscale; colour is not used as the sole carrier of meaning in any figure or table. Full PDF/UA structural tagging is not applied to this edition.

Contents

1 Introduction and Motivation

1.1 The Puzzle

A practitioner asks a frontier language model to explain the renormalization group flow of a coupling constant, and receives an answer that would pass a qualifying exam. The same practitioner, in the same session, asks how many times the letter r appears in a common fruit's name, and receives a confident wrong number. Neither result is an anomaly. Both are characteristic.

The instinct is to treat the second result as a defect that will be patched and the first as the true measure of the system. That instinct is the error this document is written to correct. The two results are produced by the same mechanism operating on inputs that differ in a property the mechanism is sensitive to and human intuition is not. Until a builder understands which property, that builder cannot predict where the system will fail—and a system whose failure surface cannot be predicted cannot be safely placed in a consequential loop.

The Dinner Table

Imagine an employee with a photographic memory for everything ever written, who has never once seen a word spelled out letter by letter. They read in whole chunks—roughly the way you read a stop sign without sounding it out. Ask them about the history of the Peloponnesian War and they are magnificent. Ask them to count the letters in "strawberry" and they are guessing, because the letters were never individually available to them. They are not stupid, and they are not being careless. The letters were never handed to them as separate things, so they have to work them out from memory—and memory is where the mistakes live. The whole discipline of building with these systems comes down to knowing which questions are the war and which are the letters.

1.2 Why This Matters Outside the Laboratory

For a consumer chatbot, an occasional miscount is a screenshot on social media. For the domains this framework targets—financial infrastructure, energy and water systems, defense logistics, regulated manufacturing, government services—the calculus is different in three specific ways.

Asymmetric loss. The cost of a wrong answer is not the mirror image of the value of a right one. A correctly summarized compliance filing saves an analyst forty minutes. An incorrectly summarized one that clears review can cost a license. Systems whose expected value is positive under a symmetric loss function can be sharply negative under a realistic one.

Auditability. Regulated environments require not merely a correct output but a defensible account of how it was produced. A system that is right for reasons it cannot expose is, in these settings, only conditionally useful.

Adversarial pressure. Consumer deployments face users who want the system to work. Infrastructure deployments face some population of users who want it to fail in a specific, profitable direction. Prompt injection, retrieval poisoning, and tool-call manipulation are not hypothetical—they are the predictable consequence of connecting a text-conditioned policy to real actuators.

These three facts push the design toward architectures amenable to verification, toward components whose failure modes can be characterized in advance, and away from monolithic reliance on a single model's judgment. That preference runs through every recommendation in this document.

1.3 Three Questions

The framework is organized around three questions, in dependency order:

1. **What is actually being combined?** Which architectural paradigms are under active development, what problem does each solve, and how do they compose? (Section 2)
2. **Why is capability so uneven?** What are the mechanical causes of the asymmetry between apparent sophistication and elementary failure, and which of them are artifacts that will be engineered away versus structural properties that will not? (Section 3)
3. **How should a builder respond?** Given the answers to the first two, what reference architecture, routing discipline, verification strategy, and evaluation regime should govern a deployment where being wrong is expensive? (Sections 4–5)

Two further questions follow from these and are treated in their own right: what the hardware substrate permits and constrains (Section 6), and what the trajectory looks like over the next three and seven years (Section 7).

1.4 Scope, Method, and Contributions

Scope. This is a synthesis and design document. It surveys published architectural work, offers a causal account of an observed phenomenon, and proposes an engineering response. It reports no new experiments. Where it makes quantitative claims about commercial systems, those claims are drawn from public sources as of July 2026 and are flagged where the underlying figures are vendor-stated rather than independently reproduced.

Method. The taxonomy in Section 2 organizes approaches by the problem each addresses rather than by the institution that published it, because institutional groupings obscure the fact that several paradigms are attacking the same constraint from different directions. The causal account in Section 3 proceeds mechanically: for each proposed cause, the test applied is whether the cause predicts the observed failure pattern more tightly than the alternatives, and whether interventions targeting that cause move the failure rate. The architecture in Section 4 is derived from those causes rather than assembled from current practice.

Contributions. Four, stated plainly:

1. A problem-indexed taxonomy of architectural paradigms with an explicit account of which combinations are complementary and which are substitutes (Section 2, Table 1 and Table 4).
2. A five-cause decomposition of capability asymmetry that separates *representational* causes, which are tractable, from *architectural* causes, which are not tractable within a fixed-depth forward pass (Section 3, Table 5).
3. A five-layer reference architecture organized around capability-aware routing and a verification gradient, with layer-level specifications and containment properties (Section 4).
4. An evaluation and acceptance regime keyed to failure classes rather than aggregate benchmark scores, including the observation that benchmark performance and deployment reliability decouple precisely in the region this document is concerned with (Section 5).
5. An account of grounding's blind spot—that attestation frequency measures propagation rather than truth—with a worked historical case, an evidence-grading scheme applied at corpus ingestion, and a metric separating evidentiary support from citation traceability (Sections 3.8 and 4.3.6, Appendix C).
6. A distinction between claims that were never grounded and claims strengthened in transmission. The second passes evidence grading, and the two compression mechanisms—

institutional and machine—amplify each other rather than offset. Section 3.8.3 adds a qualifier-attrition metric and argues that a single confident answer, stripped of its conditions, is structurally the same failure as a single authoritative spokesperson.

1.5 Relation to Prior Work

Each component of the proposed architecture has a prior literature, and the contribution claimed here is the synthesis rather than the invention of any part. Capability-aware routing draws on model cascades and router systems, and on the selective-prediction and abstention literature that predates language models by decades. Consequence-graded assurance is the standard structure of safety-integrity levels in functional safety and of risk-based assurance in regulated engineering generally; the contribution is its application to task classes within a single AI workflow rather than to whole systems. Provenance, documentation, and artifact retention build on model and dataset documentation practice and on established audit-logging discipline. Human oversight design, including automation bias and the failure of rubber-stamp review, has a large human-factors literature that the AI governance discussion frequently rediscovers. The complexity-theoretic account of what direct-answer computation cannot do, and of why intermediate token emission helps [12, 11, 13], is not original here and is cited rather than derived.

What this document adds is the joining: a specific claim that the failure surface of these systems is mechanically generated and therefore *predictable in kind if measured*, and an architecture in which the measurement (a capability registry) is wired directly into the control (routing) and the assurance (a verification gradient), with the artifact trail as the common substrate. A reader looking for novelty in any single box will not find it and should not.

1.6 How to Read This Document

Section 2 surveys the paradigms. Section 3 analyzes the asymmetry. Section 4 presents the integrated architecture. Section 5 gives implementation, evaluation, and security guidance. Section 6 examines the computational substrate, including specialized accelerators and quantum computing. Section 7 offers near- and medium-term projections with the indicators that would confirm or falsify them. Section 8 states the limitations of the analysis. Section 9 concludes. Appendices supply a glossary, a component-level reference architecture, and metric definitions.

A reader with an architecture decision to make this quarter can read Sections 4 and 5 and the *So What* boxes throughout, and lose only the reasoning. A reader who wants the reasoning should start at the beginning.

2 Architectural Paradigms for Comprehensive AI

2.1 A Problem-Indexed Taxonomy

Research programs are usually catalogued by laboratory or by model family. That grouping is convenient and analytically useless, because it conceals the more important structure: several distinct paradigms are attacking the same constraint from different directions, and several superficially similar paradigms are solving unrelated problems. Table 1 organizes the field by the constraint each family of approaches is designed to relax.

Table 1: Taxonomy of architectural paradigms, indexed by binding constraint

Family	Binding constraint	Representative approaches
Scaling & efficiency	Compute and memory per unit of capability	Mixture of experts; state-space and hybrid-attention models; sparse attention; neural architecture search; distillation and quantization
Reasoning & reliability	Quality and stability of a single inference	Chain of thought; self-consistency; tree and graph of thought; process reward models; reinforcement-trained reasoning with verifiable rewards
Modularity & composition	Task coverage beyond parametric knowledge	Tool-augmented models; retrieval-augmented generation; code execution; multi-agent systems
Grounding & embodiment	Common-sense and physical plausibility	Multimodal pretraining; vision-language-action models; simulation and robotics
Verification & structure	Guarantees rather than likelihoods	Neurosymbolic integration; constraint solvers; formal methods; typed tool contracts
Alignment & safety	Behavior under distribution shift and adversarial input	Reinforcement learning from human feedback [18]; constitutional and rule-based training [19]; mechanistic interpretability; classifier-based safeguards

The families are not alternatives. A serious deployment in 2026 combines at least four of the six. What matters is knowing which combinations are complementary, which are redundant, and which actively interfere. That is taken up in Section 2.7.

2.2 Scaling and Efficiency

2.2.1 Mixture of Experts

Mixture-of-experts (MoE) architectures decouple total parameter count from per-token compute by activating only a subset of specialized subnetworks for each input [2, 3]. Given input representation x, expert networks $E_1, \ldots, E_N$, and a learned gating network G, the layer output is

$$\mathrm{MoE}(x) \;=\; \sum_{i=1}^{N} g_i(x)\, E_i(x), \qquad g_i(x) \;=\; \frac{\exp\big(G(x)_i\big)\, \mathbf{1}\{i \in \mathcal{T}_k(x)\}}{\sum_{j \in \mathcal{T}_k(x)} \exp\big(G(x)_j\big)}, \tag{1}$$

where $\mathcal{T}_k(x)$ denotes the top-k experts selected for x. In words: G is a small learned network that scores every expert for this input; the scores are turned into weights by a softmax restricted to the k highest scorers; each selected expert E_i processes the input; and the layer's output is the weighted sum of just those k results. Only those k experts execute, so the arithmetic cost of a forward pass scales with k rather than N.

Principle — Sparse Activation

For N experts with top-k routing, per-token compute falls by a factor of approximately N/k relative to a dense model of equal parameter count, while representational capacity continues to scale with N. The saving is in arithmetic, not in memory: all N experts must remain resident or reachable, so MoE trades floating-point operations for memory capacity and interconnect bandwidth.

That trade is the entire engineering story of MoE, and it is routinely misunderstood. A sparse model with a large total parameter count is cheap to run per token and expensive to host. For a deployment with abundant accelerator memory and constrained power, this is an excellent bargain. For an edge or air-gapped deployment with a fixed memory budget, it is often a bad one. The architecture must be chosen against the deployment's actual scarce resource.

Two failure modes are specific to MoE and worth naming because they surface as puzzling behavioral inconsistency rather than as an obvious error. *Expert collapse* occurs when the gating network learns to route most traffic to a small subset of experts, wasting capacity; it is mitigated with load-balancing auxiliary losses. *Routing instability* occurs when semantically similar inputs are routed to different experts, producing outputs that differ more than the inputs do. The second is the practically important one: it means a sparse model's output variance under paraphrase can exceed a dense model's, which matters when the same query arrives twice through slightly different templates.

2.2.2 State-Space and Hybrid-Attention Models

Standard attention [1] costs $O(L^2 d)$ time and $O(L^2)$ memory in sequence length L, which is why context extension has been the dominant cost driver of the last several years. Structured state-space models (SSMs) [4, 5] replace the pairwise attention operation with a linear recurrence,

$$h_t \;=\; \bar{A}\, h_{t-1} + \bar{B}\, x_t, \qquad y_t \;=\; C\, h_t + D\, x_t, \tag{2}$$

where x_t is the input at step t, h_t the carried state, y_t the output, and $\bar{A}, \bar{B}, C, D$ are learned matrices governing respectively how much of the previous state persists, how the new input enters it, how the state is read out, and a direct input-to-output path. Read plainly, the model keeps a running summary, updates it with each new token, and answers from the summary— whereas attention re-examines every earlier token on every step. The parameters are input-dependent in the selective variants, which is what lets the model choose what to keep. The essential property is that cost scales *linearly* in sequence length rather than quadratically, and that the recurrent state carried between steps is of fixed size $O(d_{\text{state}})$—constant with respect to L, not constant in absolute terms—rather than growing with the sequence as an attention key–value cache does. Exact arithmetic cost is architecture-dependent; linear scaling in L is the defensible general claim. Training remains parallelizable through an associative scan.

The practical outcome is that SSMs are strong on long, smooth sequences and weaker on tasks requiring precise retrieval of an arbitrary earlier token—the fixed-size state is a lossy summary, and attention's key-value cache is not. This is why the architecture that has actually won deployment is neither pure attention nor pure SSM but the hybrid: interleaved recurrent and attention blocks, with the recurrence carrying long-range context cheaply and a minority of full-attention layers preserving exact lookup. Hybrid designs now appear across open-weight and commercial families, and the sparse-attention variants released through 2025 and 2026 that materially cut inference pricing are the same idea attacked from the attention side.

Recommendation — Long-context selection

Do not select a long-context architecture on advertised context length. Select it on *effective* retrieval accuracy at the position and density your workload actually uses. A model advertising a one-million-token window and a model advertising two hundred thousand may perform identically on your documents, and the cheaper one may perform better. Measure with your own corpus before the architecture is load-bearing.

2.2.3 Neural Architecture Search, Distillation, and Quantization

Three efficiency techniques matter operationally more than they do intellectually. Neural architecture search automates the discovery of topologies under a hardware-aware objective, and is used primarily to fit models to specific accelerators rather than to discover novel designs. Distillation trains a smaller student to match a larger teacher's output distribution and is now the standard route from a frontier model to a deployable one—most of the small models in production service are distilled, and the capability compression achieved is the main reason inference costs have fallen faster than hardware costs. Quantization reduces numerical precision, with four-bit formats now standard on current accelerator generations.

The caution here is specific. Distillation and aggressive quantization preserve average-case performance far better than they preserve tail behavior. A quantized student that matches its teacher within a point on aggregate benchmarks may diverge substantially on the rare, hard, or adversarial inputs that constitute the risk in a regulated deployment. Acceptance testing for a compressed model must be run on the failure classes, not on the benchmark suite.

2.3 Reasoning and Reliability

2.3.1 From Prompted Chains to Trained Reasoning

Chain-of-thought prompting [6] elicits intermediate steps before a final answer. Its mechanism is often described mystically and is in fact mundane: the forward pass has fixed depth, so a problem requiring more sequential composition than the network has layers cannot be solved in one pass. Emitting intermediate tokens converts serial depth into sequence length, and sequence length—while finite, bounded by the context window and the generation budget—is far more elastic than depth, which is fixed at training time. The model is writing to a scratchpad it can subsequently read.

Principle — Serial Depth Externalization

A constant-depth transformer is limited, under stated assumptions about precision and width, to a bounded circuit-complexity class as a function of input length, and is therefore disadvantaged on *inherently serial* task families. Emitting intermediate tokens provably extends what such a model can compute, because each generated token adds a step of sequential computation that the fixed forward pass does not supply [11, 13]. The engineering content: where a task is inherently serial, the reliable interventions are those that add sequential computation—token emission, recurrence, external memory, search, or deterministic execution—rather than a larger model of the same shape.

The precise theoretical statement matters, because a looser version of it circulates and is wrong. It is *not* established that a transformer cannot compose more dependent operations than it has

layers; layer count is not the governing quantity. What is established is a family of results, under specific assumptions about numeric precision, embedding width, and how the architecture scales with input length, placing constant-depth transformers inside bounded circuit classes, together with the complementary result that chain-of-thought strictly increases expressiveness [12, 11, 13]. Those results support a directional design rule, not a numeric budget.

This principle is the single most useful idea in the document, and most of Section 4 is its engineering consequence.

The evolution since 2024 has been from *prompting* for chains to *training* for them. Reinforcement learning against verifiable rewards— mathematical answers that can be checked, code that either passes tests or does not, formal proofs a checker accepts—produces models that generate long internal reasoning traces without being asked, and that allocate more inference compute to harder problems. The resulting test-time compute scaling is now a first-class axis alongside parameter and data scaling, and the 2026 frontier families are all organized around it, typically exposing an explicit reasoning-effort control to the caller.

Three consequences for builders follow directly. Latency becomes input-dependent and heavy-tailed, which breaks synchronous request patterns designed for fixed-cost inference. Cost becomes input-dependent, which breaks naive capacity planning. And the reasoning trace is not a faithful account of the computation—it is a generated artifact that correlates with the computation, which means it is useful for debugging and dangerous as evidence.

Caution — Reasoning traces are not audit logs

A visible reasoning trace is generated text, subject to the same failure modes as any other generated text. It may rationalize a conclusion reached by other means, omit the step that actually determined the answer, or read as sound while the answer is wrong. Treating a trace as an explanation satisfies a governance checkbox without delivering the property the checkbox exists to guarantee. Where an auditable derivation is genuinely required, it must come from a symbolic or executed artifact—a solver certificate, a query plan, an executed computation—not from prose the model produced about its own processing.

2.3.2 Search over Reasoning Structures

Chain of thought explores a single linear path. Three generalizations relax that. *Self-consistency* [7] samples multiple chains and takes a majority vote over final answers, which is the cheapest reliability improvement available and works because errors are more diverse than correct answers. *Tree of thought* [8] maintains a branching search over partial solutions with an evaluation function at each node, permitting backtracking. *Graph of thought* [9] generalizes further to permit merging and revisiting of partial results, which suits problems where subproblems recombine.

The cost scaling is the deciding factor. If a single chain costs c, then self-consistency over m samples costs mc; tree search with branching factor b and depth d costs on the order of $b^d c$ absent pruning. The value of a process reward model—a critic trained to score intermediate steps rather than final answers [10]—is precisely that it makes pruning effective, converting an exponential search into a tractable one.

Table 2: Reasoning strategies: cost, gain, and appropriate application

Strategy	Relative cost	Primary gain	Use when
Single chain	$1\times$	Depth beyond one forward pass	Default for multi-step tasks
Self-consistency	$m\times$	Variance reduction; error cancellation	Answer is short and comparable; budget permits
Tree of thought	$b^d\times$ (pruned: far less)	Backtracking from dead ends	Search-like problems with checkable partial states
Graph of thought	Variable	Subproblem reuse and merging	Decomposable problems with shared substructure
Process reward model	+critic cost	Makes pruning effective; catches mid-chain errors	Long chains where a late error is expensive
Trained reasoning (RLVR)	Input-dependent	Depth without prompt engineering	Verifiable domains: math, code, formal logic

A point that is easy to miss: every strategy in this table improves reliability on problems the model can *sometimes* solve. None of them help on problems the model cannot solve at all, and self-consistency in particular will confidently converge on a shared wrong answer when the error is systematic rather than stochastic. Voting cancels noise. It amplifies bias.

2.4 Modularity and Composition

2.4.1 Tool Augmentation

Tool use [17] is the most consequential architectural development of the period, and its importance is undersold when it is described as a convenience feature. A model that can call a calculator, execute code, query a database, or invoke a solver is not merely faster—it has acquired a different error profile. Within the tool's domain, the failure mode shifts from *plausible fabrication* to *correct execution of a possibly wrong call*. The second is far easier to detect, log, constrain, and test.

Principle — Error-Mode Substitution

The purpose of tool augmentation is not speed. It is the replacement of an unverifiable failure mode (fabrication indistinguishable from correct output) with a verifiable one (an incorrect but inspectable call and a deterministic result). Design tool interfaces to maximize this substitution: prefer typed, narrow, logged, side-effect-explicit contracts over general-purpose escape hatches.

A general code-execution tool is powerful and largely forfeits this benefit, because "the model wrote and ran arbitrary code" restores an unbounded failure surface. Where it is used, it belongs in a sandbox with no network egress, a hard timeout, and an artifact that is retained for audit.

2.4.2 Retrieval-Augmented Generation

Retrieval-augmented generation [16] separates knowledge from parameters. A query is embedded, semantically similar passages are retrieved from a corpus, and those passages are supplied in context. Formally, the generation objective becomes

$$p(y \mid q) = \sum_{z \in \mathcal{R}(q)} p_{\text{ret}}(z \mid q)\, p_{\text{gen}}(y \mid q, z), \tag{3}$$

where $\mathcal{R}(q)$ is the retrieved set. The practical benefits are substantial: knowledge updates without retraining, citation to source, corpus access control, and a large reduction in fabrication on corpus-covered questions.

The failure modes are equally specific and are where most production RAG systems actually break. Retrieval failure yields confident answers from an empty or irrelevant context. Chunking artifacts sever the sentence that carried the qualifier from the sentence that carried the claim. Embedding mismatch causes lexically dissimilar but semantically identical queries to miss. And retrieval poisoning—an adversary placing content in the corpus specifically to be retrieved—is a live attack surface wherever the corpus admits external writes.

Caution — The corpus is an attack surface

Any document that can enter the retrieval corpus can place text into the model's context. Any text in context can attempt to redirect behavior. A retrieval pipeline over a corpus with external write access is functionally an unauthenticated remote instruction channel into your system. Treat corpus ingestion with the controls you would apply to code deployment: provenance, review, signing where feasible, and the standing assumption that retrieved content is untrusted data rather than instruction.

2.4.3 Multi-Agent Composition

Multi-agent systems decompose a task across specialized model instances with defined roles—planner, executor, critic, verifier—coordinated by a protocol. The appeal is architectural: role separation, independent verification, and parallel exploration.

The evidence for the appeal is weaker than the enthusiasm. Multi-agent configurations reliably improve outcomes where the roles have genuinely different information or genuinely different tools. They improve outcomes far less where the "agents" are the same model under different instructions, since correlated errors survive the review step—a critic drawn from the same distribution as the author shares the author's blind spots. And they introduce compounding failure. The familiar way to express this—n sequential steps each succeeding with probability p give end-to-end success p^n, so 0.95 across twelve steps yields 0.54—is a useful intuition pump and a poor model of what actually happens, because it assumes step failures are independent and in agent loops they emphatically are not. A bad plan guarantees a bad execution; a misread input corrupts every step downstream. Under that positive association the true joint success rate is *higher* than p^n, since the failures concentrate in the same runs rather than spreading across them, and in the limiting case of perfect correlation the workflow succeeds at rate p regardless of n.

The correction is not reassuring, and it is important not to misread it as such. What positive correlation buys is fewer failing runs; what it costs is that the runs which fail tend to fail *completely and coherently*, producing an end-to-end result that is internally consistent and wholly wrong—the hardest class of output for a reviewer to catch. Independence would give more failures, each more visibly partial. So p^n understates reliability and overstates detectability at the same time, and the practical conclusion holds either way: bound the chain, and measure end-to-end rather than inferring from step-level numbers (Appendix C). Multi-agent designs are a reliability *multiplier* in both directions.

Recommendation — When to use multiple agents

Use role separation when the roles differ in *tools* or *data access*, not merely in prompt. Use an independent verifier only if it is independent in some real sense—a different model family, a symbolic checker, or a deterministic test. Bound chain length explicitly and require a success-probability budget: if per-step reliability times step count does not clear your acceptance threshold, shorten the chain rather than adding another reviewing agent.

2.5 Grounding, Embodiment, and Multimodality

The grounding hypothesis holds that a system trained only on text has learned the statistical shadow of the world rather than the world, and that certain classes of common-sense competence require sensorimotor contact with physical reality. Multimodal pretraining and vision-language-action models are the partial test of that hypothesis.

The honest current state is that multimodality has delivered clear gains on tasks with visual structure and has *not* resolved the asymmetry this document is about. A model with vision can often count letters in an image of a word—which is informative, since it confirms the bottleneck is representational access rather than an inability to count—while continuing to fail on the same task presented as text. That is a useful diagnostic result and a limited engineering one.

The embodiment debate remains genuinely open and should be reported as open. Whether physical grounding is *necessary* for robust common sense, or merely one efficient route to acquiring it, is not settled by current evidence, and confident claims in either direction outrun the data.

2.6 Verification and Neurosymbolic Integration

Neural systems generalize and degrade gracefully; symbolic systems guarantee and fail loudly. Neurosymbolic integration attempts to place each where it is strong. Table 3 states the division of labor.

Table 3: Complementary properties of neural and symbolic components

Property	Neural	Symbolic
Input tolerance	Handles noise, ambiguity, novelty	Requires well-formed input
Guarantees	Statistical; no correctness proof	Exact within the specified domain
Failure mode	Silent, plausible, confident	Loud, explicit, localized
Knowledge acquisition	Learned from data	Specified by humans
Auditability	Weak; post-hoc	Strong; derivation is the artifact
Coverage	Broad and uneven	Narrow and reliable
Marginal cost of new domain	Data collection	Knowledge engineering

The integration patterns worth knowing are three. *Neural front end, symbolic core*: the model translates natural language into a formal specification, a solver or checker executes it, and the model renders the result in prose. This is the pattern behind reliable natural-language interfaces to structured data, and it is the highest-assurance pattern available today. *Symbolic constraint*

on neural output: generation is constrained to a grammar, schema, or type, eliminating a whole class of malformed output. *Symbolic verification of neural proposal*: the model proposes, a checker disposes, and rejected proposals are regenerated—the pattern behind trained reasoning in formally verifiable domains.

So What

The paradigms are not competitors, and treating them as a menu is the error. The only designs that hold up under load combine a sparse or hybrid core for efficiency, trained reasoning for depth, retrieval for currency, tools for exact computation, and symbolic verification wherever an incorrect answer is expensive. Every one of these is a way of buying back a property the base architecture does not have. Know which property you are buying, and what it costs, before you add the component.

2.7 What Composes, What Conflicts

Table 4 states the interaction structure directly, because this is where architecture decisions are actually made and where most published surveys stop short.

Table 4: Composition matrix: complementary, redundant, and interfering pairings

Component A	Component B	Relation	Note
Trained reasoning	Tool use	Complementary	Reasoning decides *whether* to call; the tool supplies exactness
Trained reasoning	Self-consistency	Partly redundant	Both spend inference compute on variance reduction; stacking yields diminishing returns at high cost
Retrieval	Long context	Partly substitutes	Long context reduces but does not remove the need for retrieval; retrieval remains cheaper and auditable
Retrieval	Symbolic verification	Complementary	Retrieval supplies the premises; verification checks the inference—note this leaves premise *quality* unchecked (Section 3.8)
Retrieval	Corpus-propagated error	Interfering	Relevance ranking prefers well-attested content, so retrieval amplifies rather than corrects an error that has spread widely
Multi-agent critique	Same-model critique	Interfering	Correlated errors; creates the appearance of review without its substance
MoE sparsity	Edge deployment	Interfering	Memory-resident total parameters conflict with a fixed edge memory budget
Aggressive quantization	Tail-risk workloads	Interfering	Average-case parity conceals tail divergence
Code execution	Auditability	Conditional	Restores an unbounded failure surface unless sandboxed and artifact-logged

3 Capability Asymmetry: A Causal Analysis

3.1 Stating the Phenomenon Precisely

The informal observation is that these systems are "brilliant at hard things and stupid at easy things." That framing embeds the error. Let $D_H(t)$ denote human-perceived difficulty of task t and $A_M(t)$ denote a model's accuracy on it. For a human expert, A is approximately monotone decreasing in D_H: tasks a person finds trivial are tasks they get right. The claim this document makes about current models is that the corresponding relationship is weak:

$$\operatorname{corr}\big(D_H(t),\, 1 - A_M(t)\big) \;\ll\; 1, \tag{4}$$

and is plausibly negative over some regions of task space.

Caution — Equation (4) is a hypothesis, not a result

This document does not measure the quantity in Equation (4) and no such measurement is reported here. Testing it would require a defined task distribution, an instrument for human-perceived difficulty with established inter-rater reliability, a specified model sample and decoding protocol, an accuracy criterion appropriate to open-ended output, and an uncertainty estimate. The equation is included because it states the phenomenon precisely enough to be tested and to be argued with. It should be read as the framework's central empirical conjecture, and the framework's practical recommendations are deliberately constructed so that they remain sound whether the correlation turns out to be weak or merely unreliable.

The models are not defective on an ordering they otherwise respect. They appear to respect a different ordering. The engineering problem is that our intuitions are calibrated to D_H, so our review effort is allocated to the wrong tasks: we scrutinize the physics answer and wave through the letter count.

Principle — Calibration Transfer Failure

Human intuitions about which tasks are risky do not transfer to machine systems, because the two orderings of difficulty are generated by different mechanisms. Every review process, sampling scheme, and quality gate that allocates attention by human-perceived difficulty is therefore misallocating attention by construction. Acceptance testing must be indexed to the model's difficulty ordering, which has to be measured rather than assumed.

Five recurring mechanisms contribute to that different ordering. Table 5 summarizes them and separates those that are engineering artifacts from the one that is architectural.

Caution — These five are not an exhaustive taxonomy

They are the mechanisms this framework is built to contain, selected because each maps to a distinct architectural response. They are not a complete account of why model output is unreliable, and they mix levels of explanation— representation, data, architecture, and observer psychology. At least the following are omitted and matter in deployment: prompt sensitivity and template variance; context interference and position effects in long inputs; distribution shift between evaluation and production traffic; stochastic decoding variance; conflicting or under-specified instructions; optimization artifacts including reward hacking and sycophancy; and tool-selection error, which is a failure of the framework's own routing layer rather than of the model. A deployment's evaluation suite should cover these regardless of whether the framework names them as causes.

Table 5: Five recurring mechanisms behind capability asymmetry, and their tractability

Cause	Mechanism	Class	Tractability
Sub-token opacity	Tokenizer destroys character-level structure before the model sees it	Representational	Tractable; addressed by tool delegation, byte-level models, multimodal access
Distribution mismatch	Explicit reasoning is written down; tacit reasoning is not	Data	Partly tractable; synthetic data and RLVR shift it
Compression vs. retrieval	Parametric memory is lossy compression; interpolation is indistinguishable from recall	Representational	Tractable via retrieval grounding; not via scale alone
Bounded serial depth	Fixed-depth forward pass cannot compose beyond its layer count	Architectural	**Not tractable in-pass**; requires externalization
Anthropocentric calibration	Our surprise is misallocated, not the model's competence	Observer	Not a model property; a process defect on our side

3.2 Cause One: Sub-Token Opacity

Language models do not receive characters. They receive token identifiers produced by a subword tokenizer—byte-pair encoding or a variant—which maps character sequences to integers drawn from a fixed vocabulary. Let $T : \Sigma^* \to V^*$ be that map. The mapping is invertible—s is recoverable from $T(s)$, which is how the text is reconstructed on output—so it is not correct to say the information is destroyed. The limitation is sharper and more specific than that: character structure is not *presented* to the computation as separate addressable units. A word rendered as two or three tokens arrives as two or three atoms, and any character-level operation must be performed on representations in which the characters are not the operands.

The consequence is that character composition has to be *learned* as an association rather than *read off* as structure. Cosma et al. frame this as a low-mutual-information problem and show, across nineteen controlled character-level tasks, that this competence emerges late and abruptly in training, following the pattern of ordinary commonsense knowledge acquisition rather than that of a built-in primitive; they also show that a modest architectural change substantially improves it [15]. Interpretability work likewise finds character information is not present at the embedding layer but is progressively recovered in later layers. Learned rather than given is exactly why the errors are systematic rather than random, why they cluster on words with unusual tokenization, and why they are unreliable rather than uniformly absent.

The diagnostic that settles the causal story: the same model given the same word *as an image* frequently succeeds, and the same model instructed to insert separators between characters before counting frequently succeeds. Both interventions restore character-level access. Neither changes the model's arithmetic. The bottleneck is representational access, not counting ability.

The Dinner Table

It is the difference between reading and hearing. If someone reads you a sentence aloud, you can answer nearly anything about its meaning. You cannot reliably say how many times the letter e appeared, because you never received the letters—you received the sounds. Ask for a spelling and the person will pause and reconstruct it from memory, which is why they sometimes get it wrong. The machine's situation is the same, one level down. This is also why the fix is not to lecture the model into being more careful. The fix is to hand the question to something that can see the letters.

Byte-level and character-aware architectures dissolve this cause and pay for it in sequence length and compute. The pragmatic response in a production system is neither to wait for the architecture nor to prompt around it, but to route: send character-level queries to code, not to the model.

3.3 Cause Two: Training Distribution Mismatch

Models learn the distribution they are trained on, and that distribution is written human output, which is a badly biased sample of human cognition. Advanced material is written down carefully, at length, with the reasoning made explicit, by people being graded on the explicitness. Elementary reasoning is performed and never articulated: nobody publishes a step-by-step derivation of how they knew a coffee cup would tip when nudged, or which of two objects fits in the other.

The consequence is that the training corpus is dense precisely where human competence is rare, and sparse precisely where human competence is universal. Graduate physics is overrepresented relative to its share of human cognition by orders of magnitude. Everyday physical inference is essentially absent as *explicit reasoning*, present only as unstated presupposition.

This cause is partly tractable, and the tractability is where much of the 2024–2026 progress came from. Synthetic data generation can manufacture the missing explicit reasoning. Reinforcement learning against verifiable rewards can teach reasoning procedures without requiring a human to have written them down. Both work. Neither fully closes the gap, because generating the missing data requires knowing what is missing, and what is missing is by definition what nobody thought to write.

3.4 Cause Three: Compression Versus Retrieval

Training compresses a corpus vastly larger than the parameter count into the weights. Compression at that ratio is necessarily lossy, and the loss is not uniform: frequently-attested, mutually-consistent facts survive; rare, isolated, or contested details are reconstructed rather than recalled.

The critical property is that the model has no privileged access to which case it is in. Reconstruction and recall are the same operation from the inside. A plausible interpolation across the manifold of similar facts produces output indistinguishable, to the generating process, from retrieval of a stored one. This is the mechanism behind fabricated citations, invented statutory sections, and confidently wrong version numbers—all of which share the profile of being locally well-formed and globally false.

Principle — Indistinguishability of Interpolation

A generative model cannot reliably distinguish a retrieved fact from a plausible reconstruction, because both are produced by the same forward computation. Calibration signals derived from the model's own output—stated confidence, hedging language, self-assessment—are therefore weak evidence. Confidence must be sourced externally: from retrieval provenance, from tool verification, or from agreement across independent systems.

Note what this rules out. It rules out solving fabrication by asking the model whether it is sure. It rules out solving it by prompting for humility. Both change the surface expression of confidence without changing its relationship to correctness. The only durable fixes are architectural: ground the claim in a retrievable source, or verify it with something that can be wrong loudly.

3.5 Cause Four: Bounded Serial Depth

This is the architectural mechanism, and the one least amenable to being solved by a larger model of the same shape.

A transformer forward pass has fixed depth. Under stated assumptions about precision and width, the class of functions a constant-depth transformer can compute in a single pass as a function of input length is bounded [12, 11]. Tasks that are *inherently serial*—where each operation consumes the previous operation's output and the dependency cannot be parallelized—are the family this bound bites hardest on. Direct-answer computation on such tasks is disadvantaged relative to computation that is permitted to unroll.

Three qualifications keep this honest. Depth is not the only relevant quantity: width, precision, and how the architecture scales with input length all enter the theorems, and empirical failure on a given task may reflect training or representation rather than expressiveness. Depth does grow with model scale, slowly, so the bound moves even if it does not disappear. And architectures that are not fixed-depth—recurrent-latent designs, adaptive-computation schemes, explicit external memory—are not covered by these results at all, which is precisely why they appear in the medium-term scenarios of Section 7.

What the mechanism predicts is a signature: accuracy that is high on short dependency chains and degrades as chain length grows, with degradation steepening rather than flattening, because a single early error propagates through everything downstream. Compositionality studies report this shape [14]. Figure 1 illustrates it qualitatively.

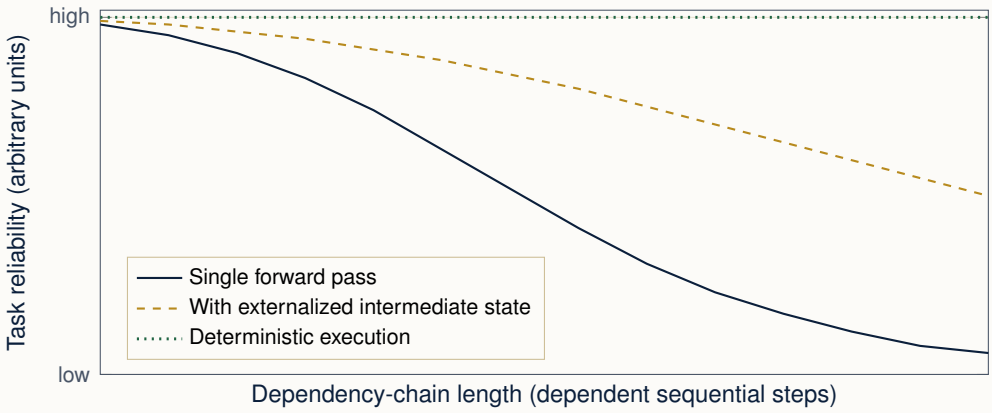

Figure 1: SCHEMATIC — NO AXIS CARRIES MEASURED VALUES. Qualitative relationship between dependency-chain length and reliability under three regimes. Axis scales are deliberately unquantified because no measurement is reported here; the figure asserts an ordering and a shape, not magnitudes. Direct single-pass computation decays fastest, externalizing intermediate state flattens the decay without eliminating it, and deterministic execution is flat by construction. The inflection point of the first curve is workflow-specific and must be measured per deployment. Figure produced by the author in TikZ.

Three regimes appear in that figure, and the gap between them is the whole engineering argument. The single-pass curve collapses. Externalizing intermediate state—emitting tokens, writing to scratch memory, maintaining an explicit structure—flattens the decay substantially but does not remove it, because each externalized step is itself performed by a fallible process and errors still compound. Handing the computation to deterministic execution is flat, because a loop does not get tired at step nine.

Recommendation — Chain-length discipline

Treat dependency-chain length as a first-class design parameter with an explicit budget. For each workflow, count the dependent steps, then *measure* where reliability on your own task classes begins to fall away; that inflection is the budget, and it is workflow-specific rather than universal. As a starting point for instrumentation—not a finding—chains beyond a handful of dependent steps are worth measuring before they are trusted. Where measurement shows decay, externalize state explicitly or delegate the composition to deterministic code. Enlarging the model of the same architecture is the least reliable of the available responses.

3.6 Cause Five: Anthropocentric Difficulty Calibration

The fifth cause is not a property of the model. It is a property of the observer, and it is the reason the first four causes keep surprising people.

Human difficulty ratings track how much deliberate effort a task takes a person, which tracks how recently the capability evolved and how little dedicated neural hardware supports it. Symbolic manipulation is effortful and recent. Visual scene parsing is effortless and ancient—and enormously more computationally demanding. Moravec's observation from the 1980s was that this ordering inverts for machines [21], and it has held up through every subsequent paradigm shift including this one.

So when a model does graduate physics and fails at letter-counting, the correct description

is not that it is inconsistent. The correct description is that our labels "hard" and "easy" were measuring the wrong quantity all along, and the machine is being entirely consistent with respect to a difficulty metric we do not natively perceive. Restated as an engineering fact: *surprise is not evidence of malfunction, and the absence of surprise is not evidence of reliability*.

3.7 What the 2025–2026 Reasoning Generation Did and Did Not Fix

The trained-reasoning generation of models is the largest capability shift since instruction tuning, and it is worth being precise about which of the five causes it addressed, because the marketing does not distinguish them.

Table 6: Effect of reinforcement-trained reasoning on each mechanism

Cause	Effect	Explanation
Sub-token opacity	Little direct effect	The tokenizer is unchanged; improvement comes indirectly, from models learning to delegate character-level work to tools
Distribution mismatch	Substantial	Verifiable rewards generate the explicit reasoning the corpus lacked
Compression vs. retrieval	Modest	Longer deliberation catches some fabrications; the indistinguishability of interpolation is untouched
Bounded serial depth	Substantial, by externalization	Extended traces are exactly the externalization mechanism; the per-pass bound is unchanged
Anthropocentric calibration	None	Not a model property

The pattern is clear and it is the pattern the framework is built on. Trained reasoning did not remove the depth bound—it made the models systematically better at working around it, which is a different and more durable achievement. The bound is still there. A reasoning model with its effort control set low is a single-pass model again, and the failure modes return with it.

Caution — Configuration is a capability boundary

Where a model exposes a reasoning-effort or thinking-budget control, that control is not a cost dial. It moves the system between capability regimes with materially different failure profiles. A workflow validated at high effort and deployed at low effort has not been validated. Pin the setting in configuration, record it in the audit trail, and re-run acceptance tests when it changes.

3.8 Grounding's Blind Spot: When the Corpus Carries the Error

Section 3 offered retrieval as the durable fix for the compression mechanism: ground the claim in a source rather than trusting parametric recall. That prescription has a boundary, and the boundary is load- bearing enough that leaving it unstated would be a defect in the framework.

Retrieval grounds a claim in what has been *written*. It does not ground it in what is *true*. Where those diverge—where an error has propagated widely through a literature—retrieval does not correct the error. It confirms it, with excellent provenance, and the verification machinery of

Section 4 reports success.

Principle — Attestation Is Not Evidence

The frequency with which a claim appears in a corpus measures its *propagation*, not its truth. Retrieval provenance establishes that a source asserted something; it does not establish that the source had grounds, or that the grounds were ever checked. A retrieval system weighted toward well-attested content is, by construction, weighted toward whatever has been copied most—which is a property of citation dynamics, not of reality.

3.8.1 A Worked Case: The Five-Micron Threshold

For most of the twentieth century, infection-control practice divided respiratory particles into "droplets" and "aerosols" at a threshold of five micrometres, with the associated doctrine that droplet-borne pathogens travelled only one to two metres. The threshold appeared in textbooks, in clinical training, in hospital protocols, and in guidance documents. It was, by any measure available to a reader, extremely well attested.

Randall and colleagues reconstructed its origin in 2021 and found no empirical study behind it [30]. Their account traces the number to an entanglement of two unrelated findings: mid-century tuberculosis research on the particle size small enough to reach the deep lung, and Wells's earlier work on the much larger size—on the order of a hundred micrometres—at which droplets fall to the ground before evaporating. The former concerned what could *infect* once inhaled; the latter concerned what could *remain airborne*. Somewhere in the citation chain the two questions merged, a threshold crystallized, and the provenance dissolved. Subsequent work has examined how the resulting dichotomy shaped the response to an airborne pandemic [33].

Three features of this case make it the right illustration for an AI framework, and none of them require any view about the public-health decisions that followed.

The error was invisible to every available verification method. A reader checking the claim would find it cited. A reader checking the citation would find another citation. Provenance was intact at every hop and the chain terminated in nothing. This is precisely the failure mode a retrieval pipeline cannot detect, because a retrieval pipeline checks that a source exists, not that the source had grounds.

It is still propagating. A survey of current sources conducted for this document found the five-micrometre threshold recited verbatim in United States patent filings issued in 2024 and 2025—years after its refutation was published in the peer-reviewed literature. Corpora do not retract. Any system retrieving over technical documents will find the error abundantly attested and the correction comparatively rare, and a relevance-ranked retriever will prefer the former on frequency alone.

The corrective scholarship models the behaviour the framework wants. Randall et al. state their central reconstruction with explicit hedging—they report that no source makes the association explicit, and that the conflation *likely* produced the threshold. They distinguish what the record shows from what they infer from it. That distinction is exactly what the original error had lost, and exactly what a well-built system must preserve.

3.8.2 The Common Mechanism

The parallel to Section 3 is structural rather than rhetorical. A generative model cannot distinguish retrieval from reconstruction because both exit the same forward pass. An institution cannot easily distinguish "we have evidence that X is false," "we have no evidence that X is true," and

"our inherited framework assumes X is false," because all three exit the pipeline as the same sentence in the same guidance document. The five-micrometre threshold was the third kind wearing the clothes of the first.

Stated generally: *confidence detaches from evidence whenever the provenance chain stops being inspectable*. In a model, compression into parameters destroys the chain. In an institution, citation lineage and guidance-document inheritance destroy it. The substrate differs; the failure is the same, and so is the remedy—make provenance inspectable, source confidence externally, and preserve the distinction between what a source showed and what was inferred from it.

3.8.3 Two Failure Modes, Not One

The five-micrometre case is one kind of attestation failure. There is a second with a different signature, and the distinction matters because the controls that catch the first do not catch the second.

Table 7: Two attestation failure modes and their controls

	Type A: never grounded	Type B: grounded, then strengthened	Control
Primary source	Does not exist; chain terminates in other assertions	Exists, and is often strong	—
Location of error	In the claim itself	In the *distance* between what the source said and what the corpus says it said	—
Under evidence grading	Caught: grades E4	**Passes**: traces to a real primary, grades E1 or E2	Grading is necessary and not sufficient
Detection	Trace the citation chain to termination	Compare the claim's stated strength against the primary's	Type A at ingestion; Type B by artifact comparison

Type B is the more common failure and, in the framework as specified so far, the undetected one. A system can trace a claim to a large, well-conducted, peer-reviewed study, assign it the highest evidence grade, and still assert something the study did not say—because what propagated through the corpus was the finding with its conditions removed.

Principle — Qualifier Attrition

Restrictions travel worse than claims. Each retransmission of a finding tends to drop conditions—the population studied, the effect size, the confidence interval, the scope—because conditions are costly to carry and add nothing to the utility of the statement for whoever is repeating it. The result is a claim that is more general, more confident, and more useful than its source, arriving with that source's authority attached. No step in the chain need be dishonest for the endpoint to be false.

This is worth stating carefully, because the intuitive framing—that someone lied—is both wrong and analytically disabling. The mechanism's significance is precisely that it produces a

falsehood without requiring dishonesty at any step, which means it recurs among entirely honest actors and cannot be controlled by selecting for integrity. Attributing it to intent also supplies an easy rebuttal: demonstrate that no individual step was dishonest and treat that as a refutation. It is not one.

3.8.4 Worked Case: A Conditioned Result Becomes an Unconditioned Rule

The Women's Health Initiative hormone-therapy trials, whose principal results appeared in 2002, are a clean instance [31]. The trial was large, randomized, and well conducted—an E1 source by any grading scheme. Its findings were also conditioned in ways stated in the publication itself: a participant population aged 50–79 with a mean near 63, well above the typical age of menopause onset; specific formulations and doses; and prespecified age subgroups.

What propagated was the unconditioned version. Hormone therapy use in the United States, above forty per cent of the relevant population in 2001, fell sharply. Subsequent analyses and extended follow-up developed what became known as the timing hypothesis—that benefit–risk varies substantially with age at initiation and time since menopause, with a materially more favourable profile for initiation before sixty or within ten years of menopause [32]—and specialty guidance moved toward individualized assessment. The qualification was restored in the literature. The unconditioned rule persisted in general circulation for roughly two decades afterwards.

Note what this case is and is not being offered as. It is not a claim about what any patient should do; current practice is individualized and this document takes no clinical position. It is a demonstration that a genuine, strong, correctly reported primary source can sit at the end of a citation chain whose endpoint asserts something the source did not, and that evidence grading alone will certify that endpoint. The gap is not in the source. It is in the transmission.

3.8.5 Why the Two Mechanisms Compose Rather Than Cancel

Institutional and machine compression are driven by different pressures, which is why they multiply.

Institutional attrition is incentive-directional. Conditions are dropped because they are expensive to carry and unrewarding to repeat, and the variant that spreads fastest is the one most useful to whoever carries it. The selection pressure is interest.

Machine attrition is attestation-directional. Hedges are low-information tokens under a predictive objective and are among the first things a lossy compressor discards. The model reproduces the modal form of a claim, and the modal form is whatever is most frequent.

These do not cancel, because the corpus a model trains on is the *output* of the institutional process. The hardened version is the frequent version— hardening is what allowed it to spread—so the model inherits an already-compressed artifact and compresses it again. The result is two-stage amplification in which the model has no access to the original condition, because the condition was lost from general circulation before training began.

One asymmetry is favourable and is the reason this is worth engineering against. Institutional attrition is effectively irreversible: the qualifier is gone from every copy in circulation, and restoring it requires re-reaching every holder. Machine attrition is potentially reversible, because the primary source frequently still exists somewhere in the corpus—it is merely outranked. Frequency-ranked retrieval will never surface it. Evidence-graded retrieval, with an explicit comparison against the primary, can.

Principle — The Single Authoritative Voice

The institutional demand for one confident spokesperson is itself an attrition generator: designate a single authority, and every condition that authority drops for brevity becomes doctrine, while the medium punishes hedging because a qualified answer reads as evasion. The role structurally produces the compression that later becomes the falsehood.

The system analogue is exact, and it is a property of the *interface* rather than of the model: a system that returns one confident answer with no visible uncertainty occupies the same structural position and generates the same failure downstream. This is the substantive justification for the Layer 1 requirement in Section 4 that user-facing uncertainty be derived from verification outcomes and displayed—not because calibration is a virtue, but because an interface that suppresses conditions is the first stage of a conflagration whose later stages nobody will be able to correct.

Caution — Inherited calibration error

A system tuned to defer to authoritative sources inherits those sources' *unstated* uncertainties as though they were settled findings. This converts an external error into an internal one that provenance-based verification structurally cannot see, because the source is authoritative by construction and the citation resolves. The verified-claim rate defined in Appendix C would score such an answer at 100%.

The response is emphatically **not** to tune the system toward scepticism of consensus. Consensus is usually correct, and a model that doubts more is differently biased rather than better calibrated; contrarianism is not calibration, and a system that questions authoritative sources at a higher rate will be wrong more often in exchange for being wrong in less familiar places. The implementable response is to represent *evidence strength* faithfully and make it inspectable, which is a metadata and verification requirement rather than a disposition. Section 4.3.6 specifies it, and Section 3.8.3 adds the second control it requires.

So What

Four of the five mechanisms are properties of how these systems represent and process input, and one is a defect in how we allocate our own attention. Only one—bounded serial depth—is structural, and it is the one that dictates architecture: any workflow whose correctness depends on a long chain of dependent steps must externalize that chain or delegate it, because no amount of model quality alone reliably converts a constant-depth direct answer into a long serial computation.

To that add the boundary condition of Section 3.8: grounding fixes what parametric memory got wrong about the corpus, and does nothing about what the corpus itself got wrong. A well-propagated error arrives with perfect provenance. So the system must carry not only *where* a claim came from but *how strongly it was ever established*—and must keep those two facts apart, because a claim can be impeccably sourced and never once tested. Design to the model's difficulty ordering, not to yours; verify what you would not have thought to check; and put the deterministic machinery where the chain is long.

4 The Integrated Development Framework

4.1 Design Principles

The architecture that follows is derived from Section 3 rather than assembled from current practice. Five principles govern it.

1. **Route by measured capability, not by apparent difficulty.** The system must hold an explicit model of what each component is reliable at, and that model must be built from measurement rather than intuition.
2. **Externalize state whenever the chain is long.** Serial depth is the structural bound; the response is to move intermediate state out of the forward pass and into inspectable storage.
3. **Grade verification to consequence.** Not everything needs a proof. Verification strength should be a monotone function of the cost of being wrong, and that function should be written down.
4. **Prefer loud failures to quiet ones.** Given a choice between a component that fails visibly and one that degrades silently, take the visible failure even at some cost in average performance.
5. **Contain, do not merely reduce.** Assume component failure at some rate and design the blast radius. A system whose safety rests on the model being right is not a safe system; it is a lucky one.

4.2 The Five-Layer Architecture

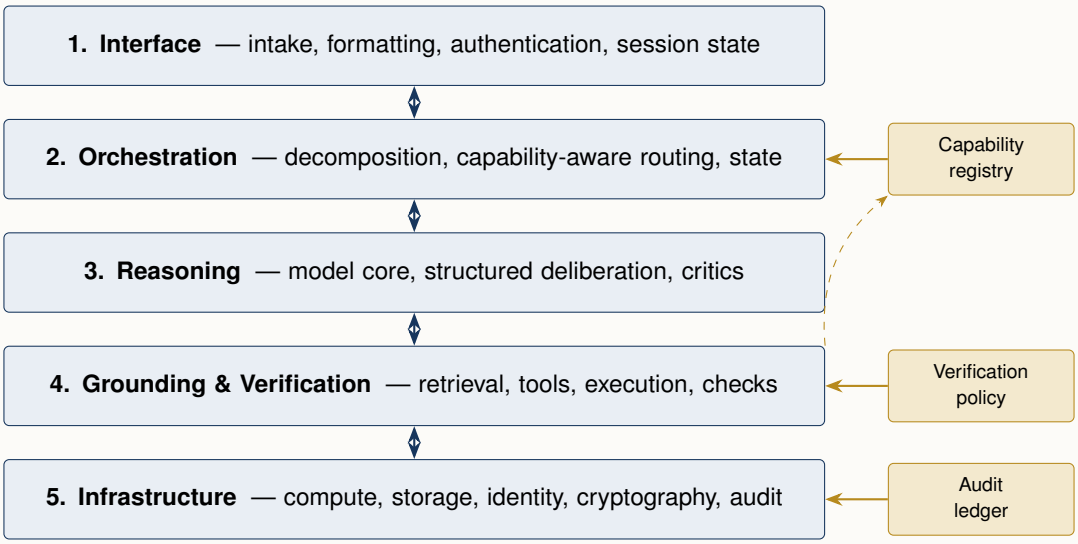

Figure 2: Five-layer reference architecture. The three gold controls are the elements that distinguish this framework from a conventional stack: a capability registry that governs routing, a verification policy graded to consequence, and an audit ledger that records what was actually executed. The dashed return path carries verification outcomes back into the capability registry, so the routing model improves from observed failures.

4.3 Layer Specifications

4.3.1 Layer 1 — Interface

Handles intake, multimodal input normalization, output formatting to domain requirements, authentication, and session state. Two design obligations are specific to this framework. First, the

interface must preserve the raw input alongside any normalized form, because normalization can destroy the very structure a downstream component needs—a document flattened to text has lost the layout that made a table a table. Second, the interface is where user-facing uncertainty is expressed, and it must express the *system's* uncertainty, derived from verification outcomes, rather than relaying the model's stated confidence, which Section 3 argues is weak evidence. This requirement carries more weight than calibration hygiene: by the single-authoritative-voice argument of Section 3.8.3, an interface that returns one confident answer with conditions stripped is the first stage of an attrition cascade, and the later stages are outside the operator's reach.

4.3.2 Layer 2 — Orchestration

Decomposes tasks, routes subtasks, and manages state across multi-step work. This is the layer where the framework's central discipline lives.

Functions: task classification; decomposition into subtasks with an explicit dependency graph; capability-aware routing; externalized state management; retry, timeout, and escalation policy; chain-length budgeting.

Principle — Capability-Aware Routing

The orchestration layer maintains a registry mapping task classes to components with measured reliability on that class. A subtask is routed to the highest-assurance component that covers it, not to the most general one. Where no component's measured reliability meets the class's threshold, the correct action is escalation to a human, not best-effort execution with a disclaimer.

The routing procedure is stated concretely because the abstraction is where implementations go wrong:

1. Classify the subtask into a class c (character-level manipulation, arithmetic, factual lookup, code generation, summarization, judgment, ...).
2. Retrieve from the registry the set of components covering c with their measured reliability $r(\cdot, c)$ and cost.
3. Retrieve the consequence tier of the enclosing workflow and its required assurance threshold τ.
4. Select the lowest-cost component with $r \geq \tau$. If none exists, escalate.
5. Execute, verify per Layer 4 policy, and write both the result and the verification outcome to the audit ledger.
6. Update $r(\cdot, c)$ from observed outcomes on a defined cadence.

Step 6 is the one most often omitted, and it is what makes the registry an asset rather than a stale document. A capability registry that is written once at design time describes the system you intended to build.

4.3.3 Layer 3 — Reasoning

Houses the model core, structured deliberation strategies, and critic components. Design obligations: the reasoning-effort setting is pinned in configuration and recorded per call; deliberation strategy is selected per task class rather than globally; critics are independent in a substantive sense (different family, different tools, or deterministic) or are not called critics.

4.3.4 Layer 4 — Grounding and Verification

Supplies external truth and checks internal claims: retrieval over governed corpora, typed tool invocation, sandboxed execution, symbolic and constraint checking. This layer implements the verification gradient.

4.3.5 Tiering is a risk assessment, not a consequence label

Consequence alone does not determine required assurance. A severe but self-announcing and instantly reversible failure may warrant less machinery than a moderate one that is silent and propagates. The tier assigned to a workflow should therefore be a function of at least six factors, not one:

Consequence. Magnitude of harm if the error reaches its effect.

Likelihood. Measured per-class reliability from the registry, on traffic resembling production—not a benchmark score.

Reversibility. Whether the action can be undone, and at what cost and latency.

Detectability. Whether the error announces itself, and how long it would persist undetected. This is the factor most often ignored and the one that drives the silent-failure rate.

Exposure. Number and vulnerability of affected parties; whether harm concentrates on people with no relationship to the deployment.

Control effectiveness. Measured, not assumed—a control that is nominally present but routinely bypassed contributes nothing.

Table 8 states the consequence axis, which is the spine of the assessment; the remaining five factors move a workflow up or down from the tier its consequence alone would suggest. The direction of adjustment is asymmetric by design: low detectability or low reversibility should raise a tier, while favourable factors should not lower one below what consequence dictates without documented justification.

Table 8: The verification gradient: consequence tier to baseline verification

Tier	Consequence of error	Required verification	Residual risk owner
0	Cosmetic; user notices and retries	None; model output is the deliverable	User
1	Wasted work; recoverable in-session	Schema validation; self-consistency as a *variance* control only	User, informed
2	Downstream propagation; recoverable at cost	Retrieval grounding with citation; independent critic	Reviewing operator
3	Financial, legal, or safety exposure	Deterministic execution or symbolic check *plus* independent validation of the specification and inputs; human sign-off on the *artifact*, not the summary, under explicit workload limits	Named accountable officer
4	Irreversible; life-safety or systemic	Model advises only; no automated action; two-person rule; full artifact retention. Note that advisory output still carries anchoring and automation-bias risk, which must be controlled separately	Governing authority

The gradient's purpose is to make an implicit decision explicit. Every deployment already assigns verification effort somehow; most assign it by intuition, which Section 3 argues is miscalibrated. Writing the tiers down forces the question of which tier a workflow is in, and that question is usually answerable even when the underlying risk is not quantifiable.

Caution — What each verification method does not establish

Every method in Table 8 has a boundary, and treating any of them as general-purpose assurance is how a gradient becomes theatre.

- **Self-consistency is not independent verification.** Samples come from one model on one prompt; it reduces stochastic variance and leaves systematic error untouched. It belongs at Tier 1 as a variance control and nowhere above.
- **A second model is not necessarily an independent critic.** Models sharing a family, training corpus, or post-training regime share blind spots. Independence must be argued from a difference in family, tools, or data access, and the argument recorded.
- **Deterministic execution validates execution, not specification.** A correctly executed query built on a wrong assumption returns a confidently wrong number with an audit trail. Tier 3 and above require the specification and its input assumptions to be checked

separately from the computation.

- **Retrieval grounding establishes provenance, not support.** A citation shows a passage was retrieved, not that it entails the claim made from it.
- **Human sign-off decays without workload control.** Automation bias is well documented; a reviewer facing high volume, low base-rate errors, and a fluent summary will approve. Sign-off requires bounded review volume, artifact-first presentation, and periodic seeded-error testing to confirm the control still functions.

Caution — Sign-off on summaries is not sign-off

At Tier 3 and above, human review of a model-generated *summary* of an artifact provides substantially less assurance than it appears to, because the summary is produced by the same process whose reliability is in question. The reviewer must be presented with the artifact—the executed query, the solver certificate, the source passage—and the interface must make reviewing it the path of least resistance. If reviewing the artifact is harder than accepting the summary, the control will decay to a rubber stamp within weeks of deployment.

4.3.6 Evidence Grading: Separating Provenance from Support

Section 3.8 established that retrieval provenance and evidentiary strength are different quantities, and that a verification layer tracking only the first will certify a well-propagated error. Closing that gap requires carrying a second attribute alongside provenance, assigned at corpus ingestion rather than inferred at query time.

Table 9: Evidence grades and permitted use by consequence tier

Grade	Meaning	Typical marker	Permitted use
E1	Direct primary evidence; result reproduced or independently replicated	Registered study with replication; measured data with method disclosed	All tiers
E2	Primary evidence, single source or unreplicated	One study; one measurement campaign; vendor data with method disclosed	To Tier 3 with the limitation stated in output
E3	Contested; credible sources disagree	Active dispute in the literature; conflicting guidance	To Tier 2; above that, the disagreement itself must be surfaced, not resolved by the system
E4	Widely asserted, original grounds not located	Claim traced only to other assertions; citation chain terminates without a primary source	To Tier 1 only; **never** the sole basis for a Tier 3 or 4 action
E5	Ungraded; provenance recorded but strength not assessed	Default for un-triaged corpus content	Tier 0–1; treated as E4 for any consequential use

Grading addresses Type A failures from Section 3.8.3 and does not address Type B: a claim strengthened in transmission traces to a genuine primary and grades E1 or E2. That case requires the attrition check specified below. With that limit stated, grade E4 is the category that matters and the one no existing pipeline records. It is where the five-micrometre threshold lived for sixty years: universally cited, never grounded. A corpus triage that asks only "is this source authoritative?" cannot produce it. The question that produces it is *"where does this claim bottom out?"*—and it is answerable, because following a citation chain to its termination is mechanical work that a system can do at ingestion and a human cannot do at query time.

Recommendation — Grade at ingestion, on the claims that matter

Grading an entire corpus is infeasible and unnecessary. Grade the claims that appear as premises in Tier 3 and Tier 4 workflows—a set that is small, enumerable, and stable. For each, trace the citation chain to termination and record the grade with the trace. Default everything else to E5 and treat E5 as E4 wherever consequence is material. The cost is bounded by the number of load-bearing premises, not by corpus size, and the artifact produced—a recorded chain terminating in a primary source or failing to—is exactly what an auditor should be shown.

4.3.7 The Attrition Check

For any claim admitted as a premise at Tier 3 or above, record two strengths: the strength asserted by the corpus statement being relied on, and the strength asserted by the primary it traces to. Where the first exceeds the second, the difference is qualifier attrition and the primary's wording governs.

This check is cheaper than it sounds, and materially cheaper than judging truth. It requires no assessment of whether the claim is correct—only a comparison of scope and hedging language between two texts, one of which the grading step has already located. It is mechanical wherever the primary is available, and where the primary is not available the claim was E4 to begin with.

Recommendation — Carry the conditions, not just the citation

When a premise is admitted, store the primary's own statement of scope— population, dose, interval, effect size, boundary conditions—alongside the citation, and require that any output relying on that premise carry the binding conditions forward. A citation without its conditions is the artifact this entire section is about. Systems that surface "*Source: [study]*" and stop have implemented provenance and omitted the part that prevents the failure.

4.3.8 Illustration: Individual-Level Inference in Clinical Settings

A domain where both failure modes bite at once is worth stating explicitly, because it is the application most often cited as AI's clearest benefit.

The argument for individual-level clinical inference is strong and largely correct: a clinician cannot integrate the volume of data a single patient now generates, population averages describe a patient nobody is, and much of the evidence base rests on trial populations that resemble the patient in front of the clinician only loosely. A system that reasons over an individual's own longitudinal data is addressing a real deficiency rather than an imagined one.

Two cautions follow directly from this framework, and neither argues against the application.

Conditioning shrinks the evidence base it draws on. The more individual features a recommendation conditions on, the smaller the effective sample supporting it, and the more the system is fitting noise rather than signal. Population averages are lossy and *robust*; individual-level inference is precise and *fragile*. The framework's own machinery applies without modification—the model cannot distinguish a recommendation reflecting a genuine subpopulation signal from one interpolated across the manifold of similar patients, exactly as Principle "Indistinguishability of Interpolation" describes. Individual-level inference done without that check is overfitting with a stethoscope.

The evidence grade of the underlying premises transfers, and is usually worse than assumed. Clinical practice inherits a large stock of E4 content: thresholds, reference ranges, and rules of thumb that are universally cited and whose original grounds are difficult to locate. A system reasoning over individual data while silently importing ungraded premises produces personalized conclusions resting on unexamined foundations—which is more dangerous than the population average it replaced, because it carries the authority of specificity.

The framework's response is the same in both cases and is not a restriction on the application: grade the premises, verify individual-level recommendations against a deterministic or statistical check where one exists, place the workflow at the tier its consequences warrant, and put the derivation in front of the clinician rather than the conclusion.

4.3.9 Layer 5 — Infrastructure

Compute, storage, identity, network isolation, cryptography, and the audit ledger. Three properties are non-negotiable in the target domains: every model call, tool invocation, and verification outcome is logged with inputs, outputs, model identity, and configuration; the cryptographic layer is agile, so primitives can be replaced without redesign (Section 5.4.8); and the execution environment for model-generated code has no network egress and hard resource bounds.

4.4 Containment Properties

The framework's claim is not that failures are eliminated. It is that failures are contained, and containment is a checkable property. Table 10 maps each cause from Section 3 to the layer that contains it.

Table 10: Mechanism-to-containment mapping

Cause	Containing layer	Mechanism
Sub-token opacity	2 (routing)	Character-level task classes route to deterministic code, never to the model
Distribution mismatch	2 (routing) + evaluation	Retrieval does *not* repair missing tacit or procedural competence; the containment is measurement—task classes where reliability is unmeasured or low are excluded from automation and routed to human fallback, with targeted or synthetic evaluation data built for the gap
Compression vs. retrieval	4 (grounding)	Retrieval's proper target: claims requiring factual support are grounded with recorded provenance or refused; no unsourced assertion above Tier 1
Bounded serial depth	2 (state)	Chain-length budget enforced; long chains externalized or delegated
Anthropocentric calibration	2, 4 (policy)	Verification is assigned by consequence tier and measured reliability, not by how hard the task looks

So What

The architecture reduces to two commitments that a reviewer can check in an afternoon. First, is there a written, measured registry of what each component is reliable at, and does routing consult it? Second, is verification strength tied to a written consequence tier, with the artifact—not a summary of it—placed in front of the human at the tiers that matter? A system with both is defensible under audit and degrades predictably. A system with neither is running on the assumption that the model is right, which is the one assumption this document exists to remove.

5 Implementation, Evaluation, and Security

5.1 Technology Selection

Table 11 maps task classes to the appropriate component. It is deliberately opinionated; a selection matrix that hedges every row is a survey, not a decision aid.

Table 11: Task class to component selection

Task class	Primary component	Rationale and caveat
Character or string manipulation	Deterministic code	Tokenizer opacity makes model performance unreliable and non-monotone; never route to the model
Arithmetic, unit conversion	Tool or code execution	Exactness is available for free; model arithmetic is interpolation
Structured data query	Model-generated query, executed	Neural front end, symbolic core; the executed query is the audit artifact
Current or proprietary facts	Retrieval with citation	Parametric memory is stale and lossy; require provenance above Tier 1
Constrained optimization	Solver, model formulates	Model translates the problem; the solver certifies the answer
Multi-step derivation	Trained reasoning + verification	Externalization is the mechanism; verify the endpoint independently
Summarization, drafting	Model core	Genuine comparative advantage; verify factual claims separately from prose quality
Judgment under ambiguity	Model advises, human decides	No verification method exists; do not automate above Tier 2
Classification at volume	Fine-tuned small model	Cheaper, faster, more stable than a frontier model; measurable on a held-out set

5.2 Implementation Patterns

Externalize state explicitly. Maintain workflow state in a typed store the orchestration layer owns, not in the conversation context. Context is a lossy, unbounded, adversary-writable medium; a typed store is none of those things.

Provide a deterministic escape hatch per task class. For every class in the registry, there should exist a non-model path—even a slow or partial one—that can be invoked when verification fails or the model is unavailable. A system with no escape hatch has a single point of failure that happens to be a vendor.

Type the tool contracts. Tools should accept and return typed, validated structures. An untyped tool interface pushes validation into prose, where it cannot be enforced.

Separate retrieval from instruction. Retrieved content enters the context in a clearly delimited region and is treated as data. The system prompt must state this and the orchestration layer must enforce it structurally, since prompt-level instruction alone is not a security control.

Log the artifact, not the narrative. The audit record contains the executed query, the tool call and result, the retrieved passage identifiers, the model identity and configuration—not the model's description of what it did.

5.3 Evaluation and Acceptance

Aggregate benchmark scores are the wrong instrument for this class of decision, for a reason worth stating precisely: benchmark performance measures central tendency on a curated distribution, while deployment risk lives in the tails of an uncurated one. The two decouple, and they decouple hardest in exactly the regime this framework targets.

Recommendation — Failure-class evaluation

Build the acceptance suite from failure classes, not from a benchmark. For each task class in the registry, assemble: (i) nominal cases; (ii) cases at the known failure boundary for that class (long dependency chains, character-level structure, rare entities, near-duplicate distractors); (iii) adversarial cases including injection attempts through every channel that reaches context; and (iv) cases drawn from your own production traffic, refreshed quarterly. Report per-class reliability with confidence intervals, and let the registry consume those numbers directly.

Four metrics carry most of the operational weight; formal definitions are in Appendix C.

Per-class reliability. Accuracy conditional on task class, with an interval. This is the quantity routing consumes.

Verified-claim rate. The fraction of factual assertions in output that are traceable to a retrieved source or tool result. Low values indicate the grounding layer is decorative.

Silent-failure rate. The fraction of errors not caught by any verification step. This is the number that matters most and the one least often measured, because measuring it requires ground truth on cases the system claimed to get right.

Escalation precision. Of the cases escalated to a human, the fraction that genuinely required human judgment. Low precision trains reviewers to approve reflexively, which destroys the control.

5.4 Security

5.4.1 Prompt Injection and the Trust Boundary

Prompt injection is not a bug class awaiting a patch. It is the direct consequence of a design in which instructions and data occupy the same channel. Every mitigation available today—delimiting, instruction hierarchies, classifier screening, provenance tagging—is a probabilistic reduction, not an elimination. The architectural response is therefore to assume injection succeeds occasionally and to bound what it can accomplish.

Principle — Authority Separation

The authority to act must not flow through the same channel as the content being processed. Tool permissions are bound to the authenticated session and the consequence tier, not inferred from anything in context. A successful injection should be able to produce a wrong answer; it must not be able to produce a privileged action.

5.4.2 Threat Model

A deployment should be able to state its threat model in one page. At minimum it enumerates: the adversary classes in scope (external opportunist, motivated external attacker, malicious insider, compromised supplier, and—distinctly— the ordinary user whose incentives diverge from the operator's); the entry points through which attacker-controlled text can reach the model, which includes every retrieval corpus, every tool result, every uploaded document, and every upstream system whose output is summarized; the assets at risk, including model outputs, tool authority,

retrieved data, credentials, and the audit record itself; and the actions the system can take that an adversary would want. Prompt injection [20] is the delivery mechanism; tool authority is the payoff. A threat model that stops at "the model might say something wrong" has not engaged with the architecture.

5.4.3 Supply Chain and Data Provenance

Model weights, adapters, embedding models, tokenizers, evaluation sets, and retrieval corpora are all supply-chain artifacts and should carry the controls that implies: verified source, integrity checking on acquisition, recorded version and provenance, and an inventory that supports answering "what changed" after an incident. Training and fine-tuning data are poisoning targets, and retrieval corpora are the cheaper target because they need no training run to take effect. Corpus ingestion warrants signed provenance and review proportional to the write-access population.

5.4.4 Least Privilege and Action Authority

Tool credentials should be scoped to the minimum action set, issued per session, time-bounded, and never resident in a prompt, a context window, or a model-accessible file. Any tool with side effects outside the system—payment, message dispatch, record modification, physical actuation—requires an approval step bound to the authenticated principal rather than inferred from the model's proposal, and the approval interface must show the action and its parameters, not a description of them.

5.4.5 Data Governance, Retention, and the Logging Tension

Corpus provenance with signed ingestion; retention and residency policy applied to context windows and audit logs as well as to databases; explicit vendor data-use terms verified in contract rather than assumed from marketing copy.

Caution — Comprehensive logging can itself be a violation

Section 4 calls for an audit ledger recording inputs, outputs, and configuration. Applied naively this conflicts with privacy law, data-minimization obligations, contractual restrictions, classification handling, and retention limits—and it creates a concentrated, high-value target containing every sensitive input the system has ever processed. The resolution is not to log less than accountability requires but to log deliberately: classify what must be retained to reconstruct a decision, retain that; store identifiers, hashes, and references rather than payloads wherever a reference suffices; apply field-level protection and access control to the ledger with the same seriousness as to the source systems; set differentiated retention by consequence tier; and confirm the design against the applicable privacy, records, and classification regimes before deployment rather than after. An audit ledger designed without this analysis is a breach waiting for an attacker to notice it.

5.4.6 Operations: Red-Teaming, Incident Response, and Retirement

Four operational controls complete the picture and are routinely deferred past the point of usefulness. *Adversarial testing* on a defined cadence, and after any change to a model, prompt, tool contract, or corpus, with findings feeding the acceptance suite rather than a report. *Incident response* with AI-specific playbooks: what to do when a model begins producing a systematic error, when injection is detected in a corpus, when a vendor deprecates a model version mid-workflow.

Rollback, which requires that model version, prompt, tool schema, and corpus snapshot are versioned together, since rolling back one without the others reproduces neither the old behavior nor the new. And *model retirement*, including revalidation when a vendor silently updates a model behind a stable endpoint—a change control problem that most governance frameworks do not yet name.

5.4.7 Standards Crosswalk

For federal and critical-infrastructure use the framework should be readable against recognized risk management guidance rather than standing alone. Table 12 maps its components to the functions of the NIST AI Risk Management Framework [22] and the Generative AI Profile [23]. The mapping is offered as an aid to adoption, not as a claim of conformity: conformity assessment against these documents is a separate exercise with its own evidence requirements.

Table 12: Crosswalk to NIST AI RMF functions

AI RMF function	Framework component	Evidence artifact
GOVERN	Consequence tiering; named residual-risk owners; model retirement and change control	Tier register with owners; change log
MAP	Task-class taxonomy; threat model; chain-length budget; escape-hatch inventory	Class taxonomy; threat model; workflow maps
MEASURE	Per-class reliability; silent-failure rate and detection recall; verified-claim and support rates; escalation precision	Acceptance suite results with intervals; seeded-error test records
MANAGE	Capability-aware routing; verification gradient; escalation and refusal; incident response and rollback	Capability registry; verification policy; audit ledger

5.4.8 Cryptographic Agility and Post-Quantum Readiness

Systems built now in critical infrastructure will still be operating when cryptographically relevant quantum computing becomes a defensible planning assumption rather than a speculative one. NIST finalized the primary post-quantum standards in August 2024: ML-KEM for key encapsulation (FIPS 203), ML-DSA for digital signatures (FIPS 204), and the hash-based SLH-DSA (FIPS 205) [24, 25, 26]. The transition timeline is set out in NIST IR 8547, which proposes deprecation of quantum-vulnerable public-key algorithms after 2030 and disallowance after 2035 [27]. Two qualifications matter for planning: IR 8547 was issued as an initial public draft in November 2024 and, as of mid-2026, had not been finalized; and it is directional NIST guidance rather than a binding standard in itself, though its dates are widely used as the federal planning baseline and are echoed in agency migration direction.

For an AI system the specific exposure is that the audit ledger, the artifact store, and the provenance signatures are the record you will rely on years from now to defend a decision made today. Their integrity guarantees must outlast the signature algorithm that created them.

Recommendation — Design for algorithm replacement

Do not attempt to select the correct primitive for a thirty-year horizon. Design so that primitives are replaceable: abstract cryptographic operations behind an interface, version every signed artifact with its algorithm identifier, maintain an inventory of where each primitive is used, and anchor long-lived integrity claims with hash-based commitments, which are far more robust to quantum attack than public-key signatures. The property to build for is agility, not a specific algorithm choice.

6 The Computational Substrate

Sections 2–5 treated algorithms and system design. This section treats the hardware those systems execute on, because in 2026 the substrate is not a background detail—it is setting the boundary conditions on what architectures are economically viable.

6.1 The Shift from Training to Inference

The defining economic change of the current period is the growing weight of inference relative to training in AI compute demand. Two developments drive it: deployment at scale, and the arrival of test-time compute scaling described in Section 2, which made a single query's cost input-dependent and, for hard queries, large.

Caution — Scope this claim before relying on it

The stronger version of this claim—that inference already consumes the majority of aggregate AI compute or AI-attributable energy—is widely repeated and not well established. Published estimates disagree, partly because they measure different things: accelerator-hours versus energy versus spend, whole industry versus one operator's fleet, and with or without the training runs amortized across a model's serving life. Treat the direction as well supported and the level as an open question, and specify the denominator before using any figure in a capacity plan or a business case.

This has a direct architectural consequence that is frequently missed. When training dominated, the binding question was how much compute could be assembled. When inference dominates, the binding questions are memory bandwidth, memory capacity, interconnect, and power—because inference is memory-bandwidth-bound in a way training is not. Architecture choices that economize on arithmetic while spending memory, MoE foremost among them, are therefore favored by the hardware trend, but only where memory is abundant.

6.2 Accelerator Classes

Table 13: Accelerator classes and their deployment fit

Class	Strength	Weakness	Fits
General GPU	Programmability; mature toolchain; handles novel architectures	Cost per inference token; power	Training; research; heterogeneous workloads
Hyperscaler ASIC	Efficiency at fixed architecture; price-performance	Rigid; captive to one cloud	High-volume stable inference
Low-latency inference silicon	Token generation speed	Narrow workload envelope	Interactive and agentic loops
Edge NPU	Power envelope; data locality; no egress	Small models only	Air-gapped, tactical, privacy-bound deployment
Neuromorphic	Energy efficiency on sparse, event-driven workloads	Framework incompatibility; niche	Research; specific sensing applications

The structural story through 2026—*vendor-reported and analyst-projected; not independently reproduced*—is the rise of custom silicon against general-purpose GPUs for inference workloads. Every major hyperscaler now fields in-house accelerators, growth in that segment materially outpaces GPU growth, and the incumbent has responded with rack-scale co-designed platforms pairing next-generation GPUs with matched CPUs, interconnect, and high-bandwidth memory. Independent verification of vendor performance claims lags the announcements by quarters, and figures should be treated accordingly.

For the deployments this framework targets, the practical implications are narrower than the headlines suggest:

- **Portability is a procurement requirement.** Custom silicon is cloud-captive. If your architecture cannot move between accelerator families without redesign, your negotiating position degrades every year.
- **Edge capability has crossed a threshold.** Distillation plus four-bit quantization plus edge NPUs makes genuinely useful local inference feasible in air-gapped and tactical settings. The tail-risk caveat from Section 2 applies with full force.
- **Memory capacity gates architecture choice.** A sparse model's total parameters must be resident. This is the single most common mismatch between an architecture chosen on benchmark performance and a deployment environment chosen on other grounds.

6.3 Power as the Binding Constraint

The constraint that determines whether planned capacity actually arrives is firm power delivered on schedule—interconnection queues, transmission, transformers, and permitting—rather than chips or capital. Projections of data-centre electricity demand identify AI as a principal driver [29], and the direction of that finding is robust even where specific magnitudes are contested. Silicon lead times are measured in quarters; interconnection and transmission in years. For an organization planning AI infrastructure on a multi-year horizon, power availability at the intended site is

the assumption most likely to be wrong and the one least often tested early.

6.4 Quantum Computing: What It Will and Will Not Do

Quantum computing occupies a peculiar position in AI discourse, simultaneously overstated as an imminent transformation of machine learning and understated as a live cryptographic threat. Both errors are worth correcting.

6.4.1 Current State

Error correction crossed from demonstration to engineering during 2024–2026. The load-bearing published result is below-threshold operation of a surface-code *memory*: a distance-7 code on 101 physical qubits whose logical error rate fell as code distance increased, establishing that scaling reduces rather than amplifies error [28]. Since then several vendors have reported error-corrected logical qubit counts in the tens to low hundreds, decoding latencies have fallen sharply, and published roadmaps target a few hundred logical qubits from roughly ten thousand physical ones toward the end of this decade, with much larger systems in the 2030s. Resource estimates for factoring RSA-2048 have also fallen substantially from earlier figures as algorithms and codes improved—a more consequential trend than any single qubit-count announcement.

Caution — Six different claims are reported under one word

Cross-vendor comparison of "logical qubits" is unreliable because the term is used for materially different achievements. Before treating any announcement as progress toward a cryptographically relevant machine, establish which of these it is:

(i) a logical *memory* that preserves a state below threshold, with no computation performed on it—the category of the result cited above;

(ii) error-*detected* encoded qubits, where faults are flagged and the run discarded, rather than corrected;

(iii) genuinely error-*corrected* logical qubits;

(iv) logical *gates* between encoded qubits, including the non-Clifford operations that universality requires;

(v) fault-tolerant *computational* qubits sustaining deep circuits;

(vi) vendor *roadmap targets*, which are statements of intent.

The distance between (i) and (v) is most of the remaining engineering, and headline counts routinely conflate them.

6.4.2 Quantum for AI: A Sober Assessment

The honest answer is that quantum computing is unlikely to matter for mainstream machine learning on the planning horizon of this framework, for reasons that are structural rather than incidental.

- **Data loading is the bottleneck.** Encoding a large classical dataset into quantum states can cost as much as the classical computation it was meant to accelerate. Most proposed quantum machine learning speedups assume this problem away.
- **The workload is a poor match.** Neural network training is dominated by dense linear algebra at low precision on very large data sets, precisely the workload classical accelerators are optimized for and quantum devices are not.

- **The credible near-term applications lie elsewhere.** Quantum simulation of quantum systems, meaning chemistry and materials, is the domain with a genuine theoretical advantage, and it is adjacent to AI through data generation rather than through model training.

The plausible integration pattern is hybrid: a classical system, quite possibly an AI system, formulates and orchestrates; a quantum processor executes a narrow subroutine; classical infrastructure handles error correction and result interpretation. That is a specialized accelerator relationship, not a paradigm replacement, and it fits cleanly into Layer 4 of the framework as one more tool behind a typed contract.

6.4.3 Quantum Against Cryptography: The Real Exposure

The asymmetry deserves emphasis. Quantum computing's near-term significance for an AI infrastructure program is almost entirely on the cryptographic side, not the capability side. Data captured today under classical public-key protection can be decrypted later; signatures relied on today can be forged later against records that still matter. Both are relevant to an AI system precisely because its audit ledger and artifact store are long-lived integrity claims. Section 5.4.8 states the response.

So What

Budget attention proportionally. Specialized inference silicon and power availability will change your cost structure and your deployment options within the current planning cycle, and both deserve engineering time now. Quantum computing will change your cryptography before it changes your models—so treat it as a cryptographic agility program with a long fuse, not as a capability bet. The organization that gets this allocation backwards spends its scarce technical attention on the thing that will not arrive and neglects the thing already constraining it.

7 Prognosis

Projections are stated with the indicators that would confirm or refute them, because a forecast without a falsification condition is an opinion with a date attached.

7.1 Near Term: 2026–2029

Capability asymmetry narrows without disappearing. Tool delegation becomes reliable enough that character-level and arithmetic failures largely vanish from user-visible behavior—not because the underlying limitation was removed, but because routing became competent. The depth bound persists and continues to manifest on long unverifiable chains.

The frontier fragments into families rather than flagships. The pattern already visible in 2026—model families differentiated by cost, latency, and reasoning effort rather than a single flagship—continues, and selection shifts from "which model is best" to "which model fits this task class," which is the routing discipline of Section 4 arriving as a market structure.

Open-weight models remain roughly a generation behind at a small fraction of the cost, which makes self-hosted deployment the default for sovereignty-constrained and air-gapped work.

Verification becomes the differentiator. As raw capability commoditizes, competitive advantage in regulated deployment shifts to the surrounding machinery: grounding, verification, audit,

and containment. Vendors will increasingly sell this rather than model quality.

Governance formalizes around auditability. Regulatory attention converges on traceability and human accountability rather than on model internals, which favors architectures that produce artifacts.

7.2 Medium Term: 2029–2033

Three scenarios, with rough subjective weights.

Table 14: Medium-term scenarios and their leading indicators

Scenario	Weight	Description	Leading indicator
Integration	∼55%	Continued architectural integration; asymmetry managed rather than solved; systems become reliable through composition	Verification tooling matures faster than base models
Architectural break	∼25%	A new architecture dissolves the depth bound—adaptive-depth, recurrent-latent, or explicit-memory designs reaching frontier scale	A non-fixed-depth architecture appears at frontier capability, not just in research
Plateau	∼20%	Scaling and test-time compute both saturate; progress becomes incremental and application-driven	Successive frontier releases show narrowing deltas on held-out, uncontaminated evaluations

The framework in Section 4 is deliberately robust across all three. Under integration it is the correct design. Under an architectural break it remains correct and some of its machinery becomes unnecessary—a good failure. Under plateau it becomes more valuable, because when the model stops improving, everything around it must.

7.3 What Would Falsify This Analysis

Three observations would require substantial revision:

1. A fixed-depth architecture demonstrating reliable performance on dependency chains far beyond its layer count *without* externalizing intermediate state. This would refute the depth-bound argument that Section 3 treats as structural.
2. Model-reported confidence becoming a well-calibrated predictor of correctness across distribution shift. This would refute the indistinguishability-of-interpolation principle and would make much of the verification gradient unnecessary.
3. Capability asymmetry disappearing under scale alone—models becoming uniformly reliable across the human difficulty ordering without architectural intervention. This would refute the framework's premise outright.

None of the three has been observed as of July 2026. All three are the right things to watch.

8 Limitations

The analysis has boundaries, and stating them is a condition of being taken seriously.

No new empirical work, and one specific gap that matters most. This is synthesis and design. The causal account in Section 3 is consistent with published evidence and with the diagnostic interventions described, but it is not established here by controlled experiment. The sharpest form of the gap is this: *the framework has not been shown to reduce the silent-failure rate relative to a monolithic baseline*. Every recommendation here argues from mechanism to design, and mechanism-to-design arguments are exactly the kind that survive scrutiny and fail measurement. The decisive experiment is neither expensive nor exotic—instrument one workflow both ways, with and without capability-aware routing and the verification gradient, and compare severity-weighted silent-failure rates on production-like traffic—and until someone runs it, the architecture in Section 4 is a well-motivated hypothesis about how to build, not a demonstrated result.

Vendor-reported figures. Quantitative claims about commercial systems and hardware reflect public reporting as of July 2026. Independent verification lags announcements, benchmark contamination is a persistent and under-measured problem, and definitions—particularly of "logical qubit" and of context length—vary between vendors in ways that make cross-comparison unreliable.

Illustrative figures. Figure 1 is schematic. It represents a qualitative relationship, not measurement.

Rapid obsolescence of specifics. The architectural analysis and the five causes should age well; they concern mechanisms. Model names, prices, benchmark standings, and accelerator specifications will not age well and should be re-verified before any of them becomes load-bearing in a decision.

Scenario weights are subjective. The weights in Table 14 express a considered judgment, not a calculation, and they should be treated as a starting point for disagreement rather than a result.

9 Conclusions

Three findings and one instruction.

First: the asymmetry is mechanical, and four of its five causes are tractable. Sub-token opacity, distribution mismatch, and the compression–retrieval tradeoff yield to engineering—routing, synthetic data and verifiable-reward training, and retrieval grounding respectively. The fifth, anthropocentric calibration, is not a model property at all but a defect in how we allocate scrutiny, and it is fixed by writing down consequence tiers instead of trusting intuition.

Second: bounded serial depth is the architectural mechanism, and it dictates the design. Constant-depth direct-answer computation is disadvantaged on inherently serial tasks, and every effective response—chain of thought, trained reasoning, scratchpads, tool delegation, externalized state—is the same move: adding sequential computation the forward pass does not supply. This is why the framework is organized around externalization and delegation rather than around model selection. It is a directional rule grounded in expressiveness results, not a numeric law, and the budget it implies must be measured per workflow.

Third: grounding has a blind spot, and it is not a small one. Retrieval fixes what parametric memory got wrong about the corpus and does nothing about what the corpus got wrong. Attestation frequency measures propagation, not truth, so a well-spread error arrives with perfect provenance and satisfies every provenance-based control—and a system tuned to defer to authoritative sources inherits their unstated uncertainties as settled findings. The answer is not to make the system more sceptical, which trades familiar errors for unfamiliar ones. It is to grade

evidence at ingestion for the premises that carry weight, and to measure evidentiary support separately from citation traceability, so that "well-sourced" and "well-established" stop being the same number.

And grading alone is not enough, because there are two failures and it catches one. A claim can trace to a genuine, strong primary and still assert what that primary did not, because conditions travel worse than claims—and the corpus a model learns from is the output of that attrition, so the machine compresses an already-compressed artifact. Carry the primary's conditions forward with the citation, measure the gap where the corpus asserts more than the source did, and recognize that an interface returning one confident answer with the conditions stripped is not merely poorly calibrated. It is the first stage of a cascade whose later stages nobody will be positioned to correct.

Fourth: the paradigms compose, and knowing what each one buys is the skill. Sparse and hybrid architectures buy efficiency; trained reasoning buys depth; retrieval buys currency and provenance; tools buy exactness and a better error mode; symbolic verification buys guarantees. Each purchase has a price, and several combinations interfere. Table 4 is the part of this document most likely to save a reader a quarter of wasted work.

The instruction: build the capability registry and the verification gradient before you build anything else, and make the artifact—not the summary of the artifact—the thing a human reviews at the tiers where being wrong is expensive. Everything else in this framework is elaboration on those two sentences. Applied to the framework itself, the standing test: will someone examining this design a century from now conclude it was built by people who understood what they were deploying, or by people who were impressed by it? The registry, the gradient, and the artifact trail are what separate the two answers.

So What

These systems are not unreliable in a general way that patience will cure. They are reliable and unreliable in a specific, measurable pattern that does not match human intuition about difficulty. An organization that measures that pattern, routes to it, verifies in proportion to consequence, and keeps the artifacts can deploy AI in consequential settings with a defensible risk position. An organization that trusts the impressive answer because the answer was impressive has taken on a liability it has not measured and cannot bound. The difference between those two organizations is not the model they licensed. It is roughly six weeks of unglamorous engineering, done before the deployment rather than after the incident.

A Glossary

ASIC Application-Specific Integrated Circuit. Fixed-function silicon optimized for a narrow workload; efficient and inflexible.

BPE Byte-Pair Encoding. A subword tokenization algorithm that iteratively merges frequent adjacent byte pairs; the source of sub-token opacity.

Capability registry In this framework, the measured mapping from task class to component reliability that governs routing.

CoT Chain of Thought. Emission of intermediate reasoning steps before a final answer; mechanically, the externalization of serial depth.

Consequence tier A written classification of a workflow by the cost of an error, used to set required verification strength.

GoT Graph of Thought. A reasoning structure permitting branching, merging, and revisiting of partial results.

HBM High-Bandwidth Memory. Stacked memory adjacent to an accelerator die; the usual binding constraint on inference throughput.

Logical qubit An error-corrected qubit encoded across many physical qubits. Vendor definitions vary materially.

MoE Mixture of Experts. Sparse activation of a subset of expert subnetworks per token.

NPU Neural Processing Unit. Low-power inference accelerator, typically for on-device use.

PQC Post-Quantum Cryptography. Algorithms designed to resist attack by a cryptographically relevant quantum computer.

PRM Process Reward Model. A critic trained to score intermediate reasoning steps rather than final answers, enabling effective search pruning.

QEC Quantum Error Correction. Encoding logical qubits redundantly to detect and correct physical errors.

RAG Retrieval-Augmented Generation. Supplying retrieved passages in context to ground generation in a governed corpus.

RLHF Reinforcement Learning from Human Feedback. Post-training alignment from human preference comparisons.

RLVR Reinforcement Learning from Verifiable Rewards. Training against automatically checkable outcomes—proofs, tests, computed answers.

Serial depth The number of dependent sequential operations a computation requires; bounded within a single fixed-depth forward pass.

Silent failure An incorrect output not caught by any verification step; the primary risk quantity in this framework.

SSM State-Space Model. A linear-recurrence sequence architecture with constant per-step state, used to reduce the quadratic cost of attention.

Test-time compute scaling Improving output quality by spending more inference computation on harder inputs.

ToT Tree of Thought. Branching search over partial solutions with evaluation and backtracking.

Verification gradient The mapping from consequence tier to required verification method used by Layer 4 of the framework.

B Reference Architecture — Component Checklist

A deployment implementing this framework should be able to point at a concrete artifact for each row. Absence of an artifact is the finding.

Layer	Component	Required artifact
1	Input normalization	Raw input retained alongside normalized form
1	Uncertainty presentation	System-derived confidence, not model-stated
2	Task classifier	Class taxonomy with inter-rater agreement measured
2	Capability registry	Per-class reliability with intervals and refresh date
2	Chain-length budget	Documented dependency-step limit per workflow
2	State store	Typed schema; state not held in conversation context
3	Model configuration	Pinned effort setting, versioned, logged per call
3	Critic policy	Independence justification per critic
4	Retrieval corpus	Provenance, ingestion controls, write-access inventory
4	Tool contracts	Typed schemas; side effects declared; permissions bound to session
4	Execution sandbox	No network egress; resource and time bounds
4	Verification policy	Consequence tier per workflow; method per tier
4	Evidence grading	Graded premise register for Tier 3–4 workflows, with citation traces terminating in a primary source or recorded as not doing so
5	Audit ledger	Inputs, outputs, model identity, config, verification outcome
5	Cryptographic inventory	Primitive usage map; algorithm identifiers on signed artifacts
5	Escape hatch	Non-model path documented and exercised per task class

C Metric Definitions

Let $\mathcal{C}$ be the set of task classes, T_c the evaluation set for class $c \in \mathcal{C}$, $\hat{y}(t)$ the system output on case t, $y(t)$ the ground truth, and $v(t) \in \{0, 1\}$ the indicator that some verification step flagged the case.

Per-class reliability. Let $\mathrm{acc}(t) \in \{0, 1\}$ be the acceptance criterion appropriate to class c:

$$r(c) \;=\; \frac{1}{|T_c|} \sum_{t \in T_c} \mathrm{acc}(t). \tag{5}$$

Two things must be specified before this number means anything. First, $\mathrm{acc}(\cdot)$ is exact equality only for closed-form classes— arithmetic, extraction, classification. For open-ended output it must be a declared rubric with measured inter-rater reliability, or a task-specific automatic criterion whose agreement with human judgment has itself been measured; substituting exact match for a rubric silently redefines the metric. Second, the interval is binomial only when cases are independent. Evaluation sets built from templates, from multiple items per document, or from repeated sampling of the same input are clustered, and a naive binomial interval will be too narrow; use a cluster-robust or hierarchical interval, or report the effective sample size. A point estimate without an interval, or an interval computed under an assumption the evaluation set violates, should not be entered into the registry.

Silent-failure rate. The fraction of cases that are wrong *and* unflagged:

$$\mathrm{SFR}(c) \;=\; \frac{\big|\{t \in T_c : \hat{y}(t) \neq y(t) \;\wedge\; v(t) = 0\}\big|}{|T_c|}. \tag{6}$$

Note that SFR can be reduced either by improving accuracy or by improving detection, and that the second is usually cheaper. This is the primary risk metric for tiers 2 and above, and it should be reported in two refinements. *Severity-weighted*: with harm weight $w(t)$ drawn from the consequence tiering, $\mathrm{SFR}_w = \sum w(t)\mathbf{1}\{\cdot\} / \sum w(t)$, since one undetected Tier-3 error is not equivalent to twenty undetected Tier-1 errors. And *paired with error-detection recall*, $\Pr(v = 1 \mid \hat{y} \neq y)$, which is what actually measures whether the verification layer works; a low silent-failure rate achieved through high accuracy on an easy evaluation set tells you nothing about detection.

Verified-claim rate. For output containing factual assertions $a_1, \ldots, a_m$, with $\pi(a_i) = 1$ when a_i is traceable to a retrieved source or tool result. Traceability is a necessary and clearly insufficient condition: it records that a source was retrieved, not that the source *supports* the claim. Where the distinction matters—which is everywhere above Tier 2—pair this with a sampled entailment check, human or automated, and report the support rate alongside the traceability rate:

$$\mathrm{VCR} \;=\; \frac{1}{m} \sum_{i=1}^{m} \pi(a_i). \tag{7}$$

Evidentiary support rate. The verified-claim rate above measures traceability into the corpus. It says nothing about whether the corpus had grounds, which Section 3.8 shows is a separate and sometimes opposite quantity. Let $g(a_i) \in \{\mathrm{E1}, \ldots, \mathrm{E5}\}$ be the evidence grade of the strongest source supporting assertion a_i, and let τ_g be the minimum grade admissible at the workflow's consequence tier (Table 9). Then

$$\mathrm{ESR} \;=\; \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{g(a_i) \preceq \tau_g\}, \tag{8}$$

where $\preceq$ orders grades from strongest to weakest. Report ESR alongside VCR, never instead of it, and treat a large gap between them as the diagnostic it is: VCR ≈ 1 with low ESR describes a

system that is citing diligently and grounding poorly—well-sourced claims resting on premises nobody ever established. That is the signature of the failure in Section 3.8, and it is invisible to every other metric in this appendix.

Qualifier attrition. Let $\sigma(\cdot)$ map a statement of claim c to an ordinal strength level—for example, from weakest to strongest: *consistent with*, *suggests*, *shows in population P*, *establishes*, *holds generally*. Let $\sigma_{\mathrm{P}}(c)$ be the strength asserted by the primary source and $\hat{\sigma}(c)$ the strength asserted by the corpus statement relied on. Then

$$A(c) \;=\; \hat{\sigma}(c) - \sigma_{\mathrm{P}}(c), \tag{9}$$

positive when the corpus asserts more than the source did. Report the distribution of A over the graded premise register, not a mean: the operative quantity is the count of premises with $A > 0$ admitted at Tier 3 or above, which should be zero by policy. Two properties make this practical. It is defined only where a primary exists, which is exactly where evidence grading has already done the locating work; and it requires comparing scope and hedging between two texts rather than adjudicating truth, so it can be assessed without domain expertise in the claim's subject matter.

Escalation precision. With E the set of escalated cases and $h(t) = 1$ when human judgment was genuinely required:

$$\mathrm{EP} \;=\; \frac{1}{|E|} \sum_{t \in E} h(t). \tag{10}$$

Low escalation precision is a leading indicator of reviewer habituation: when most escalations turn out not to have needed a human, approval becomes the default response and the control decays while continuing to appear in the process documentation. No general threshold is defensible—the point at which this happens depends on review volume, time pressure, base rates, and interface design—so the operative practice is to track the metric over time against a baseline established for the specific deployment, and to confirm the control empirically with periodic seeded errors rather than inferring its health from a ratio.

End-to-end chain reliability. For a workflow of n sequential steps with per-step reliability p_i and independent failures:

$$R_{\mathrm{chain}} \;=\; \prod_{i=1}^{n} p_i. \tag{11}$$

This expression is *exact only under independence*, and it is neither an upper nor a lower bound in general. Dependence moves joint reliability in either direction: if step correctness is positively associated—as it is when consecutive steps share a model, a prompt template, or an input—then $\Pr(\text{all correct}) \geq \prod_i p_i$, and in the limit of perfect correlation the workflow succeeds at rate $\min_i p_i$ rather than $\prod_i p_i$. Negative association pushes it the other way. The practical consequence is that $\prod_i p_i$ should be used as a *planning heuristic for chain-length discipline*, not as an estimate of workflow reliability. Joint reliability must be measured end-to-end on realistic traffic, or estimated with an explicit dependence model; a chain-level number computed from step-level numbers is not evidence about the chain.

References

Entries are labelled by source status: [PEER] peer-reviewed; [PRE] preprint; [STD] standard or government guidance; [ORG] institutional report. Claims in the text that rest on vendor announcement, market

projection, or the author's judgment are marked as such at the point of use and are not given bibliography entries, because they are not sources in the sense this list implies.

References

[1] [PEER] Vaswani, A., et al. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems 30*.

[2] [PEER] Shazeer, N., et al. (2017). Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *International Conference on Learning Representations*.

[3] [PEER] Fedus, W., Zoph, B., and Shazeer, N. (2022). Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *Journal of Machine Learning Research* 23(120).

[4] [PEER] Gu, A., Goel, K., and Ré, C. (2022). Efficiently Modeling Long Sequences with Structured State Spaces. *International Conference on Learning Representations*.

[5] [PRE] Gu, A., and Dao, T. (2023). Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752.

[6] [PEER] Wei, J., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems 35*.

[7] [PEER] Wang, X., et al. (2023). Self-Consistency Improves Chain of Thought Reasoning in Language Models. *International Conference on Learning Representations*.

[8] [PEER] Yao, S., et al. (2023). Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *Advances in Neural Information Processing Systems 36*.

[9] [PEER] Besta, M., et al. (2024). Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *AAAI Conference on Artificial Intelligence*.

[10] [PEER] Lightman, H., et al. (2024). Let's Verify Step by Step. *International Conference on Learning Representations*.

[11] [PEER] Li, Z., Liu, H., Zhou, D., and Ma, T. (2024). Chain of Thought Empowers Transformers to Solve Inherently Serial Problems. *International Conference on Learning Representations*. arXiv:2402.12875. *Load-bearing for the depth argument in Section 3.*

[12] [PEER] Merrill, W., and Sabharwal, A. (2023). A Logic for Expressing Log-Precision Transformers. *Advances in Neural Information Processing Systems 36*.

[13] [PEER] Merrill, W., and Sabharwal, A. (2024). The Expressive Power of Transformers with Chain of Thought. *International Conference on Learning Representations*.

[14] [PEER] Dziri, N., et al. (2023). Faith and Fate: Limits of Transformers on Compositionality. *Advances in Neural Information Processing Systems 36*.

[15] [PEER] Cosma, A., Ruseti, S., Radoi, E., and Dascalu, M. (2025). The Strawberry Problem: Emergence of Character-level Understanding in Tokenized Language Models. *Proceedings of EMNLP 2025*. arXiv:2505.14172. *Load-bearing for the tokenization argument in Section 3.*

[16] [PEER] Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems 33*.

[17] [PEER] Schick, T., et al. (2023). Toolformer: Language Models Can Teach Themselves to Use Tools. *Advances in Neural Information Processing Systems 36*.

[18] [PEER] Ouyang, L., et al. (2022). Training Language Models to Follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems 35*.

[19] [PRE] Bai, Y., et al. (2022). Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073.

[20] [PEER] Greshake, K., et al. (2023). Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. *ACM Workshop on Artificial Intelligence and Security*.

[21] [ORG] Moravec, H. (1988). *Mind Children: The Future of Robot and Human Intelligence*. Harvard University Press.

[22] [STD] National Institute of Standards and Technology (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1.

[23] [STD] National Institute of Standards and Technology (2024). *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*. NIST AI 600-1.

[24] [STD] NIST (2024). *FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard* (ML-KEM).

[25] [STD] NIST (2024). *FIPS 204: Module-Lattice-Based Digital Signature Standard* (ML-DSA).

[26] [STD] NIST (2024). *FIPS 205: Stateless Hash-Based Digital Signature Standard* (SLH-DSA).

[27] [STD] Moody, D., Perlner, R., Regenscheid, A., Robinson, A., and Cooper, D. (2024). *Transition to Post-Quantum Cryptography Standards*. NIST IR 8547, initial public draft, November 2024. *Not finalized as of mid-2026; see Section 5.4.8.*

[28] [PEER] Google Quantum AI and Collaborators (2025). Quantum Error Correction Below the Surface Code Threshold. *Nature* 638, 920–926 (published online December 2024).

[29] [ORG] International Energy Agency (2025). *Energy and AI*. IEA, Paris.

[30] [PEER] Randall, K., Ewing, E. T., Marr, L. C., Jimenez, J. L., and Bourouiba, L. (2021). How Did We Get Here: What Are Droplets and Aerosols and How Far Do They Go? A Historical Perspective on the Transmission of Respiratory Infectious Diseases. *Interface Focus* 11(6): 20210049. *Load-bearing for the worked case in Section 3.8; the authors state their central reconstruction with explicit hedging, which the case discussion preserves.*

[31] [PEER] Writing Group for the Women's Health Initiative Investigators (Rossouw, J. E., et al.) (2002). Risks and Benefits of Estrogen Plus Progestin in Healthy Postmenopausal Women: Principal Results from the Women's Health Initiative Randomized Controlled Trial. *JAMA* 288(3): 321–333.

[32] [PEER] Manson, J. E., et al. (2013). Menopausal Hormone Therapy and Health Outcomes During the Intervention and Extended Poststopping Phases of the Women's Health Initiative Randomized Trials. *JAMA* 310(13): 1353–1368. *Cited in Section 3.8.3 for the restoration of conditions, not for any clinical recommendation.*

[33] [PEER] Jimenez, J. L., et al. (2022). What Were the Historical Reasons for the Resistance to Recognizing Airborne Transmission During the COVID-19 Pandemic? *Indoor Air* 32(8): e13070.

Currency note. Bibliography verified July 2026. Statements in the text about model families, pricing, benchmark standings, accelerator specifications, and vendor logical-qubit counts derive from public reporting current at that date, are identified in situ, and should be re-verified before use in any decision.

QUANTUM RESERVE PRESS · TECHNICAL FRAMEWORK SERIES

First Edition · July 2026