june 2026

Designing Trustworthy AI

Why high-stakes systems are converging on verification.

whitepaper

Designing Trustworthy AI

Why high-stakes systems are converging on verification.

author Nick Lamb, PhD

version June 2026

topic Verification, evidence, and trustworthy AI

audience AI researchers, technical operators, healthcare and pharma leaders, enterprise decision-makers

An argument shaped by direct experience building verification-first systems for medical writers, patient communication, and pharmaceutical compliance review. Sketched from the practice; not from the survey.

Contents

1

ch. 01

Executive Summary

— Fluency is not evidence.

Large language models (LLMs) are becoming remarkably capable. Whether they are becoming trustworthy is a different question, and one that is not answered by progress on the first.

The two are often treated as a single problem at different stages: capability now, reliability later. They are not. They are different design problems, governed by different constraints, and in the domains where AI is being asked to inform consequential decisions – clinical, legal, financial, scientific – they are pulling apart rather than converging.

The argument of this whitepaper is that trustworthy AI in high-stakes domains increasingly depends on system design rather than model capability alone. The systems we will trust to inform a clinical decision, vet a legal claim, summarise a regulatory filing, or interpret a scientific result will not be the ones that generate the most fluent answer. They will be the ones designed, end to end, around the assumption that generation is not enough.

The argument is shaped, in part, by direct experience building verification-first systems in regulated medical communications, where the shape of the trust problem becomes visible early.

We call this shift the **generation → verification shift**, and the systems it produces **Verification AI**.

The trust problem is not a prompting problem

The errors that matter in high-stakes settings have a particular character. They are not random noise. They are structurally predictable consequences of how generative systems work. Citations are fabricated because the model is optimising for the shape of a citation, not for its existence.1 Rationales are confabulated because explanation tokens are produced by the same probability machinery as everything else; the explanation is generated, not retrieved. Confidence is uniform because the model does not natively distinguish between claims it has strong evidence for and claims it is plausibly inventing.

These are not bugs that better prompting reliably fixes. Prompt engineering raises the floor; it does not raise the ceiling on what a generative architecture can know about its own outputs. A practitioner can reduce hallucination rates by guiding a model toward source-grounded answers, but cannot, through prompting alone, give the model a faithful internal signal for the difference between a fact and an artefact.

Larger models help, sometimes considerably. But scale changes the texture of the failures more than it reduces them. As models get better at sounding right, the gap between sounding right and being right becomes more consequential, because reviewers – including expert reviewers – calibrate to fluency. A subtly wrong answer in confident prose is harder to flag than an obviously wrong answer in clumsy prose. This is one of the quieter consequences of capability gains, and one of the least addressed.

Where authority lives

The defining question of any AI system is where its authority lives.

In a generation-first system, authority sits inside the model. Whatever the model produces is the answer; retrieval, citation and formatting exist primarily to contextualise that answer. The reader's trust, when it is given, is trust in the model.

In a verification-first system, authority sits outside the model. The model is a draftsman, not an oracle. Its outputs are candidates, not conclusions, and the structures around it – evidence, retrieval, comparison, contradiction detection, bounded scope, audit trails – decide which candidates earn the right to be shown to a user. Trust is assigned by the surrounding system, not asserted by the model.

Verification AI is the name we give to systems built on this second posture. It refers to systems that constrain, ground, audit and evaluate model outputs against external evidence rather than treating generation itself as the source of authority.

This is not a single technique. It is a design discipline that runs through five recurring principles, each of which the paper develops in its own section. *Grounded retrieval* insists that claims be tied to specific source material rather than to the model's parametric memory. *Bounded scope* limits the questions a system is willing to answer to those it is actually equipped to answer well. *Evidence comparison* checks generated claims against retrieved material before they reach the user. *Contradiction detection* surfaces, rather than smooths over, places where evidence disagrees with itself or with the draft. And *human auditability* treats the reviewer's ability to trace a claim back to its basis as a first-class property of the system, not an afterthought.

None of these are new in isolation. What is new is the willingness to treat them as the architecture, rather than as guardrails added after the fact.

Healthcare as stress test

Healthcare is not the focus of this paper because it matters more than other domains. It is the focus because reliability failures surface there sooner, more visibly and with less ambiguity than almost anywhere else.

It is evidence-rich: almost every meaningful clinical claim can, in principle, be traced to a source. It is ambiguity-rich: the evidence is often partial, contested or context-dependent, which means a trustworthy system has to do more than fetch the right document. It is stakes-rich: errors have consequences that are easy to measure and impossible to ignore. And it is institutionally cautious: clinicians, regulators and patients are not predisposed to extend trust to systems that cannot explain themselves.

A merely fluent generative system fails quickly in this environment. A system that retrieves, compares, flags disagreement, and tells the user what it does not know fails more slowly, and in more recoverable ways. Healthcare functions as a stress test for trustworthy AI: the design choices behind a system become legible sooner there, because the cost of getting them wrong arrives sooner.

The paper draws on experience building such systems: what worked, what did not, what trade-offs surfaced only in deployment, without treating healthcare as the destination of the argument.

Why this generalises

The reason healthcare is a stress test rather than a niche is that the architectural pattern travels. Anywhere a system produces claims that someone else will rely on, the same structural question arises: is the authority in the model, or in the evidence?

Finance has its own version. A model summarising an earnings release, flagging a covenant breach, or interpreting a regulatory filing faces the same generation-vs-verification choice, with the same downstream consequences when the choice is made carelessly.

Legal reasoning has it acutely. The most public LLM failures of recent years have been legal: confidently cited cases that did not exist. The pattern is not a quirk of one jurisdiction or one model. It is what happens when a generative system is asked to play a verification role it was not designed for.

Scientific publishing has it in a quieter but more consequential form, as model-assisted writing enters peer review and the line between drafted and verified content becomes harder to police. Enterprise knowledge systems have it whenever an internal assistant is trusted to answer questions whose answers will inform a decision. Policy will have it more and more as AI systems are used to interpret regulation, draft analysis and brief decision-makers.

In each case the architectural answer rhymes: the system that earns trust is not the one with the largest model, but the one with the most disciplined relationship to evidence.

The obvious objection

The obvious objection is that models are improving rapidly, and capabilities that looked out of reach 2 years ago are routine today. Why design elaborate verification architectures for a problem that successive model generations may simply outgrow?

The honest answer is that some of it they will. Hallucination rates are falling. Tool use is becoming more competent.2 Retrieval-augmented systems are getting better at staying close to sources. It would be unwise to argue that none of the current verification scaffolding will be absorbed into the models themselves.

But three things suggest the architectural question will not go away.

First, capability and calibration improve at different rates. A more capable model is not automatically a model that knows more accurately what it does not know. The gap between knowing and knowing-that-you-know has narrowed unevenly, and remains the place where high-stakes failures occur.

Second, as outputs become more persuasive, the cost of remaining errors rises. A fluent, well-reasoned, beautifully cited wrong answer is a harder failure mode than an obviously wrong one. Capability gains can shift the distribution of errors toward the ones that are hardest to catch.

Third, trust is a property of systems, not of components. Even if a future model were individually reliable, the institutions deploying it will still need auditable, bounded, evidence-grounded systems around it to satisfy regulators, reviewers and users. The verification layer is doing work that the model, however capable, cannot do on its own.

These are not reasons for pessimism about model progress. They are reasons to take seriously the parts of the trust problem that progress alone will not solve.

What the whitepaper covers

The remainder of the whitepaper develops the argument in three sections. The first examines why generation-first systems struggle in high-stakes settings, with attention to the failure modes that recur across domains. The second introduces Verification AI as a design philosophy and walks through the five principles named above, with concrete grounding in healthcare deployments and analogues in other fields. The third turns outward – to finance, law, scientific publishing, enterprise systems and policy – and asks what the verification turn implies for how trustworthy AI gets built, regulated and reasoned about over the next few years.

The argument is not that generation does not matter. It does. The argument is that, in the domains we care about most, the question of whether to trust an answer has become inseparable from the question of how the system that produced it was designed.

Capability will continue to advance, and many of the rough edges visible today will be smoother in two years. But progress on capability does not, by itself, build the structures that decide which of a model's outputs deserve to be trusted. Those structures have to be designed.

The future of trustworthy AI may depend less on whether models become universally reliable, and more on whether the systems around them are honest about what they can and cannot verify. The missing layer is not intelligence, it is architecture. That is the work ahead, and the subject of the rest of this whitepaper.

2

ch. 02

The Trust Problem

— What sounds right and what is right are different problems.

It is tempting to describe LLM failures in high-stakes settings as a long tail of unrelated errors: bad luck on individual examples, fixable by patching the model or sharpening the prompt. They are not. Spend enough time debugging these systems and the errors begin to sort themselves into a small number of recurring families. The families are stable across models, vendors and use cases. They are also stable across capability levels: better models change the rate at which the failures occur, not the shape of the failures themselves.

That stability matters. It tells us these failures are not accidents but the predictable output of a particular architectural choice – the one this whitepaper is concerned with. They are what happens when a generative system is asked to do work that requires verification, in domains where ‘mostly right’ is the wrong target.

The danger in high-stakes settings is not, in the end, that LLMs are wrong. It is that they are convincingly wrong, in patterns that human reviewers are poorly equipped to catch.

The shape of authority

Start with the most photogenic failure: the hallucinated citation. The case law that does not exist. The journal article with a real-looking identifier and an entirely invented abstract. The footnote pointing to a paper the author never wrote.

These failures are familiar enough now to have become a recognisable pattern. They are also, structurally, the easiest to explain. A generative model produces tokens that are likely under its distribution. Citations have a recognisable shape (ie, author surname, year, volume, page range) and the model is good at producing things in that shape. Existence is not part of the optimisation. The model is generating the appearance of authority, not its substance.3

In the language of the previous section: the citation is generated, not retrieved.

The more interesting cousin of the fabricated citation is the confidently weak claim. The model produces a statement with the cadence and register of expert prose – measured, qualified, technical – while the underlying support is thin or absent. The reader's eye moves through it without resistance, because nothing in the surface signals weakness. A poorly supported claim in clumsy prose draws a second look. The same claim in confident prose tends to move past.

In high-stakes review this asymmetry is the central problem. Reviewers, including expert reviewers, read for fluency cues first and evidence cues second. A draft that reads like a paragraph from a textbook is treated, by default, like a paragraph from a textbook. Generative systems are very good at producing textbook prose. They are not, by the same mechanism, producing textbook evidence.

Fluency is often mistaken for evidence. That mistake grows more dangerous as models get better.

The almost-right answer

If fabricated citations are the failure mode that makes the news, semantic drift is the one that does most of the actual damage.

A drug is approved for treatment of a condition in a specific patient population. A model, asked to summarise the indication, produces a statement that is approximately correct but quietly broader – eliding the population restriction, or rephrasing a contraindication as a relative caution, or smoothing a hedged regulatory statement into a flatter one. None of the words are individually wrong. The sentence is wrong as a sentence.

In practice, this pattern is often the most consequential failure mode in deployment, and the one least appreciated outside teams that build these systems. Fabrications can be caught with a citation check. Drift cannot. Drift is what happens when a model is doing exactly what it is supposed to do – reading source material and producing prose about it – and the prose, as prose, is well-formed. The error is not in the facts. It is in the shape of the claim.

A related pattern shows up under ambiguity. Real evidence in high-stakes domains is often partial, contested or context-dependent. A trustworthy answer in those conditions is restrained: it qualifies, attributes, sometimes refuses. Generative systems lean the other way. They are trained to produce answers; they have no internal economy that rewards saying less than they have.4 Faced with conflicting sources, the easiest output is a single confident synthesis that quietly resolves the conflict the reader was supposed to see.

This is one of the deeper sources of the trust problem: high-stakes domains reward restraint; generative systems reward completion.

The failures that emerge from this mismatch are not blatant. They are almost-right. The dosage is plausible but the population is wrong. The case holds, but not in the cited jurisdiction. The clinical evidence is real, but the strength of the recommendation has been gently upgraded in the summary. These are the errors that practitioners spend most of their time on, precisely because they are the ones a quick reading would miss.

Why we miss it

There is a tendency to discuss these failure modes as if the problem were entirely the model. Much of it lives in the reader.

Humans calibrate to surface cues. Coherent prose, professional formatting, citation-shaped strings, the right technical register. All of these raise the implicit credibility of a piece of text. They function as low-cost evidence that the author knew what they were doing. We have spent a long time using these cues in human-authored material, where they are statistically informative. In LLM output they are not.

The cues are decoupled. A model that produces a well-formatted, well-cited, well-reasoned-looking paragraph is not, by virtue of producing it, more likely to be correct. But the reader's prior is calibrated on a world where those things tend to go together, and the prior is hard to override. Automation bias does the rest:5 a confident output from a competent-looking system is trusted more than the same content would be trusted from a less authoritative source.

This is what makes capability gains an ambivalent answer to the trust problem. A more capable model produces output that looks more like the human-authored material the reader's intuitions were trained on. The cues are stronger; the cues are also less correlated with truth. A failure becomes harder to notice as the surface around it improves.

None of this is news to anyone who has worked with these systems for a while. But it is worth saying clearly: trustworthy AI is not only a property of the model. It is a property of the system the model sits inside, including the human who reads its output.

Humans make mistakes too

There is an honest objection to all of this. Human experts also make mistakes. They cite the wrong case, misread a guideline or round a hedged statement into a confident one. The failure modes described above are not exclusive to AI systems; they are, in many respects, recognisable from the human professions these systems are being asked to assist.

The problem is not perfection.

The problem is the combination of properties. Human experts make mistakes, but they make them at human scale, at human speed, with human accountability, and inside social structures that are designed to surface and correct them. A misdiagnosis is anchored to a clinician whose record is reviewable. A bad legal brief has a lawyer's name on it. A misquoted study attracts a correction notice. The systems we have built around expert work – peer review, malpractice, audit, regulatory inspection – exist precisely because individuals are fallible, and they assume that fallibility is bounded in ways that can be policed.

Generative systems do not, by default, sit inside these structures. The same fluent confidence is produced at industrial scale and machine speed, without any natural mechanism for accountability or correction. A clinician who errs once a year is one kind of risk. A system that errs at the same per-instance rate but is invoked a million times a day is a different kind. The errors are individually familiar; the aggregate is not.

There is also a more subtle asymmetry. Human experts have an internal sense of uncertainty that is, at least in healthy practice, communicated through how they speak. 'I think' and 'I'm not sure' and 'let me check' are not weaknesses; they are signals other humans rely on to calibrate trust. Generative systems can be prompted to imitate this register, but the underlying calibration is largely absent. The model that says 'I'm not sure' is, often, no less confident in its internals than the model that says 'this is clearly the case.' The expression of uncertainty has been generated, not measured.

This is not an argument that humans are better than models. It is an argument that the systems around humans were built with human failure modes in mind, and that the systems around generative AI mostly have not been.

A structural problem

What the patterns above have in common is that they are not bugs. They are the predictable output of a particular architectural choice: that the model itself is the source of authority for whatever it produces. Under that choice, fabrications, confident weak claims, semantic drift, false certainty and the asymmetry between fluency and evidence are not accidental. They are what the architecture is doing.

That framing matters because it determines what counts as a fix. If the failures were random, we would expect them to fall as models improve. They are not random: they are the consequence of asking a generative system to do verification work it was not designed for, and they appear with similar shape across model generations, prompting strategies and deployment patterns.

If these failures are structural rather than accidental, the natural next question is whether more capability (bigger models, better tools, sharper retrieval, more sophisticated prompting) eventually closes the gap.

That is what the next section examines. The question is no longer whether generative systems can improve. They clearly will. The more important question is whether improvement alone changes where trust should live.

3

ch. 03

The False Promise of Better Generation

— Hedging is a register, not a measurement.

The strongest argument against everything in this whitepaper is that future model generations may make the verification scaffolding described here look like a workaround for problems that no longer exist. Hallucinations are falling. Reasoning is sharpening. Retrieval is becoming more reliable. The most familiar verification techniques of 2025 – careful prompt engineering, system prompts that ask the model to cite sources, evaluation pipelines built around catching obvious failure modes – have already been quietly absorbed, in part, by more capable models. If that absorption continues, why design an architectural alternative?

This is the right objection to take seriously, and the place to start.

Improvements over the last two model generations have been genuine. Reasoning chains are longer and more coherent. Tool use, once brittle, is becoming a reliable substrate for agentic work. Instruction following is markedly tighter. Hallucination rates on simple factual queries are lower than they were 18 months ago. Frontier models can summarise long documents, write competent code, follow multistep plans and recover from errors in ways that would have looked aspirational 2 years ago. Anyone working with these systems daily can feel the change. None of this paper's argument depends on denying it.

The question is whether those improvements, extrapolated forward, eliminate the architectural trust problem described in the previous section, or whether they leave it largely intact while changing its shape. The answer this paper offers is the second. Better generation reduces the volume of obvious errors and the burden of obvious scaffolding. It does not, by itself, supply the structures that high-stakes work requires. It is solving the right problem for many things, and the wrong problem for these.

Why prompting is not architecture

Of all the techniques that have absorbed verification-shaped work into the model, prompting is the most overstated.

Prompting genuinely improves outputs. Asking a model to think step by step, to cite its sources, to refuse when uncertain, to consider counterarguments – these moves all measurably raise the quality of what comes back. None of this is disputed. The mistake is to treat the technique as a substitute for the architecture it imitates.

A prompt can ask a system to be careful. It cannot create the mechanism by which carefulness is enforced. It can ask for citations. It cannot guarantee that the citations exist. It can ask for hedging where evidence is weak. It cannot give the model an independent signal for how weak the evidence actually is. The instructions describe the desired behaviour; the system is still using the same generative machinery to satisfy them, which means the same failure modes – fabrication, drift, false certainty – are still in play, just hidden behind a more compliant register.

This becomes most visible at the edge cases. A well-prompted model will, most of the time, behave as the prompt asks. The trouble is that high-stakes failures concentrate exactly in the cases where it does not: when the question is ambiguous, when the source material conflicts, when the right answer is to refuse. The prompt cannot tell the model that this is one of those cases, because the model has no internal mechanism that distinguishes “this question is hard” from “this question is easy.”

“Please be accurate” is not an architecture. It is a request inside an architecture. And the underlying architecture, as long as it remains generation-first, preserves the same failure modes under any prompt.

None of this is a complaint about prompting as a discipline. Prompt engineering is genuinely useful. It raises the floor on what generative systems do well. It does not, on its own, raise the ceiling on what they can be trusted to do.

What scale changes, and what it doesn't

The argument for scale is more interesting, because it has been right so often.

Capabilities that looked architecturally hard a few years ago have turned out to be properties that emerged with more data and more parameters. Reasoning is the canonical example: an ability we once thought might require special-purpose systems became something a sufficiently large language model could do reasonably well. It is fair to ask whether reliability is the next thing on that list – whether the trust problem is just another property scale eventually delivers.

The honest answer is that scale has been moving the failures rather than removing them. Hallucination rates on simple factual prompts are lower than they were. The shape of the remaining failures has shifted toward harder cases: longer documents, more context, more ambiguous questions, more subtle reasoning chains.6 The failure modes of the previous section (drift, confidently weak claims, false certainty under ambiguity) remain recognisable in the most capable frontier models, even if their frequency and presentation have changed. They look different. They sit inside more elaborate scaffolding, wrapped in more competent-looking prose. Per instance, they are not less consequential.

There is a second-order effect that matters here. As models get better, the average human reviewer's calibration of when to trust them gets worse – because most outputs are correct, and the surface around the incorrect ones improves at the same rate as the surface around the correct ones. The cost of a remaining error rises while the rate of remaining errors falls. The product of those two trends is not obviously favourable.

This is not an argument against scale. Scale is the single most important variable in modern AI, and it has paid off repeatedly. It is an argument that the relationship between scale and trustworthiness is more complicated than the relationship between scale and capability. Capability rises smoothly. Trustworthiness rises in jagged ways, with new failure modes opening up exactly where the old ones close.

The calibration ceiling

Underneath the scale argument sits a quieter one, which is that capability and calibration are different problems with different ceilings.

Capability is what the model can do when it tries. Calibration is how reliably the model's expressed confidence tracks the actual evidence behind a claim. Capability has improved enormously. Calibration has improved more slowly, in less measurable ways, and with no clear roadmap for the kind of step changes that capability has seen.

The reason is mechanical. A generative model produces tokens; expressing uncertainty is just a different distribution over tokens. A model that has been trained to sound hedged when its evidence is thin will sound hedged where its training distribution suggested hedging is appropriate, not where the actual evidence is thin.7 The two often correlate, which is why hedging-trained models behave better. They do not always correlate, which is why a confident-sounding wrong answer can still come from a careful-sounding system.

For low-stakes work, this gap is unimportant. For high-stakes work, it is most of the problem. A clinician deciding whether to trust a summary, a lawyer deciding whether to cite a precedent, a financial analyst deciding whether to act on a flagged anomaly – all of them need a faithful signal of how confident the underlying system is, not a stylistic approximation of one.

A model that knows more is not automatically a model that knows when to hesitate. That is the calibration ceiling, and it sits well below the capability ceiling. Until they converge – and there is no reliable evidence that capability gains alone are closing the gap on the timescales people often assume – the architectural problem of distinguishing the model's expressed confidence from the actual strength of its evidence remains.

What institutions ask for

There is a final move in the optimist case that bypasses the technical arguments entirely: even if today's systems are imperfectly calibrated, the trajectory is steep enough that institutional concerns will catch up to a much more reliable substrate. Verification work will become unnecessary because the underlying systems will be reliable enough not to need it.

This argument underestimates institutions.

The infrastructure that high-stakes domains build around expert work (peer review, audit trails, signed-off recommendations, regulatory inspection, professional accountability) is not primarily a workaround for individual fallibility. It is a mechanism for making reliability legible. A clinician's competence is not measured by the percentage of correct calls in the abstract. It is measured by the fact that decisions can be traced, reviewed, challenged and corrected by other competent people who can see the basis for them. The same logic applies to legal opinions, scientific publications, financial disclosures and policy analysis. What these systems demand is not only that the answer be right. They demand that the answer be checkable.

This is the property no amount of model capability supplies on its own. A perfectly reliable oracle that produces answers without traceable bases is, from an institutional point of view, still not a trustworthy substitute for the work it replaces. The institutions cannot certify what they cannot inspect. The reviewers cannot review what they cannot trace. The auditors cannot audit what they cannot see.

Verification AI is, in part, an attempt to take this seriously. The point of grounding, retrieval, comparison, contradiction detection and bounded scope is not only that they reduce errors – though they do – but that they make the production of an answer legible to the people who will need to stand behind it. Capability gains do not produce that legibility; architecture does.

A possible future, and the question that survives it

It is worth stating the optimistic case in its strongest form, because the rest of this paper depends on taking it seriously.

It is entirely possible that future model generations narrow the calibration gap, internalise much of the current verification scaffolding and arrive at a substrate that is more straightforwardly trustworthy for many tasks. Some of the techniques described in this paper will, in that future, have been absorbed into the models themselves. The boundary between “what the model does” and “what the surrounding system does” will move. It has already moved in this direction over the last several years.

What does not move is the underlying question. Even in that future, the institutions deploying these systems will still need to know, at the moment a high-stakes claim is produced, what evidence stands behind it, what evidence contradicts it, what was checked, what was assumed and how confident the system has any right to be. That question is structural. It does not collapse into capability. A model that is internally more reliable still has to make its reliability visible to the people who depend on it.

If better generation is insufficient for that work, the question shifts. It is no longer “How do we make models generate better answers?” It is “How do we design systems that decide when an answer deserves trust?”

That is the question the next section takes up.

4

ch. 04

The Shift — From Generation to Verification

— Verification begins where generation ends.

Across the high-stakes domains where AI is being asked to do serious work, an architectural transition appears to be underway. It is quieter than the capability story that dominates public discussion of these systems. It does not announce itself in benchmark scores or model releases. But it is visible in the choices that practitioners are making about how trustworthy systems should be built – choices that have begun, across very different teams and domains, to rhyme.

The transition is from generation-first systems to verification-first systems. The previous sections have set out why this transition is likely necessary. This section is about what the transition actually is.

It is, on the surface, a change in architecture. Beneath that, it is a change in posture. The shift is not really about adding components: retrieval layers, evaluation mechanisms and comparison steps. It is about answering a different question. A generation-first system asks: can the model produce the answer? A verification-first system asks: what structures determine whether the answer deserves to be trusted?

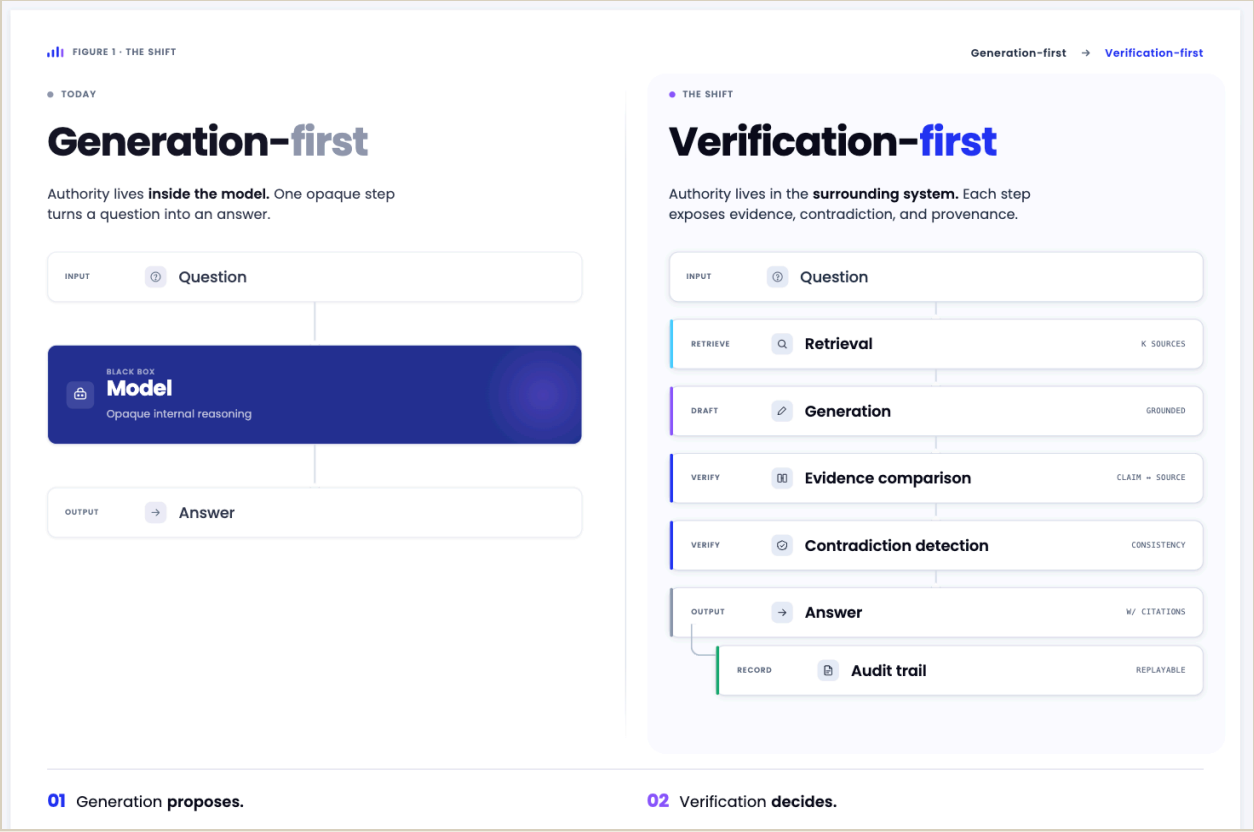

FIG. 1 The shift from generation-first to verification-first AI. In generation-first systems, authority lives inside the model. In verification-first systems, authority moves into the surrounding architecture: retrieval, evidence comparison, contradiction detection, provenance, and auditability.

The change of question is the point. Once it is asked seriously, much of the design follows.

What Verification AI is

Verification AI refers to systems that constrain, ground, audit and evaluate model outputs against external evidence rather than treating generation itself as the source of authority.

That definition deserves unpacking, because it is trying to do several things at once.

First, Verification AI is not a model. It is not a prompting technique. It is not a single architecture, and it does not correspond to any one benchmark. It is a design philosophy – a way of organising the relationship between generation and trust. The model is still doing the heavy lifting; what changes is the role assigned to the model's output. In a generation-first system, that output is the answer. In a verification-first system, it is a candidate. The candidate becomes an answer only after it has passed through structures the model itself does not control.

Second, the verification work is external. It does not depend on the model knowing what it does not know, or on the model expressing uncertainty faithfully. Those properties remain useful where present, but they are not load-bearing. Load-bearing is the surrounding system: the retrieval that brought back evidence, the comparison that checked the draft against the evidence, the contradiction layer that surfaced disagreements, the bounded scope that refused questions outside the system's competence, the audit trail that recorded what was checked and why.

Third, and most importantly, Verification AI does not eliminate generation. It demotes it. Generation becomes a powerful but provisional first move, the draft. The system decides whether the draft holds up. The model drafts; the system decides.

That last line is worth resisting as a slogan. It is true enough to use, but only if the components doing the deciding are real. The rest of this section and the next are about what real means here.

Where authority moves

The conceptual spine of the shift is where authority lives. The first section of this paper introduced the distinction; it is worth returning to with more weight now.

In a generation-first system, authority lives inside the model. The model is the source of truth for whatever it produces, and the surrounding architecture exists to deliver, present and contextualise that truth. When the system is asked a question, the implicit chain of trust is short: the user trusts the system, the system trusts the model, the model is the answer. The chain has one consequential link.

In a verification-first system, authority lives outside the model. The surrounding architecture is the source of truth – evidence, retrieved with discipline; comparisons, performed against that evidence; disagreements, surfaced rather than smoothed; questions, refused when they fall outside what the system can support. The model is one component inside that architecture, not the architecture itself. The user trusts the system because the system can show its work.

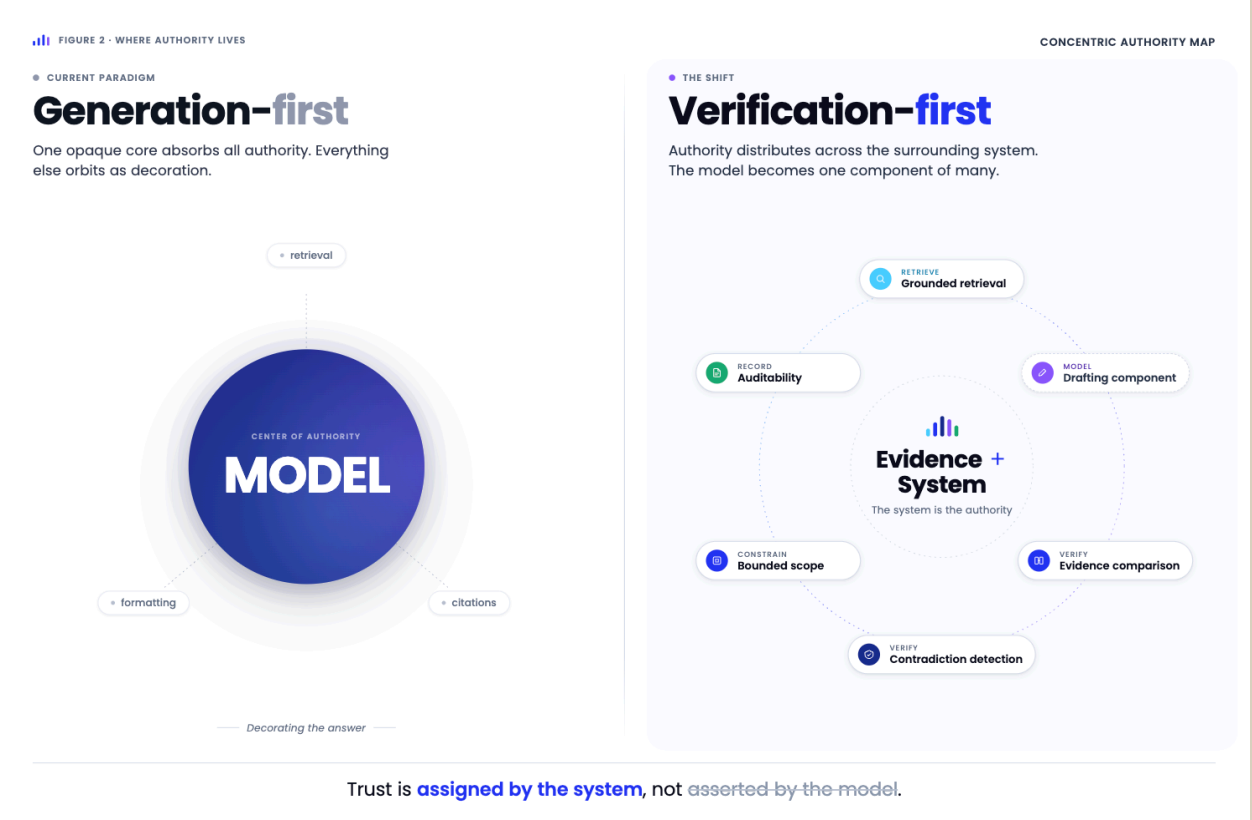

FIG. 2 Authority shifts from a concentrated model-centric architecture to a distributed verification system. In generation-first systems, retrieval, citations, and formatting orbit the model. In verification-first systems, trust emerges from interactions among retrieval, evidence comparison, contradiction detection, bounded scope, and auditability, with the model acting as one component rather than the sole authority.

The change is small to describe and large in consequence. Almost everything about how a trustworthy system gets built – what gets logged, what gets shown to a reviewer, what counts as a refusal, what counts as an answer – flows from where the authority was placed.

The reframe also changes the question the system is trying to answer. A generation-first system, in effect, asks: what answer sounds right? A verification-first system asks: what answer survives inspection? The first question has been the central engineering challenge of LLMs for the better part of a decade. The second is the one we have spent less time on, and it is the one high-stakes work is increasingly asking us to solve.

A spectrum of verification

It would be a mistake to read all of this as a binary, with pure generation on one side and Verification AI on the other. The honest picture is a spectrum, and most useful systems sit somewhere along it.

At one end is unguarded generation: a model takes a question, produces an answer, and that is the system. Chat interfaces, used loosely, sit here. The model is doing the work; nothing surrounds it; the user supplies any verification themselves, mostly by deciding whether to believe what they read.

A first layer of structure adds prompting and guardrails. The model is asked to think carefully, cite sources, defer when uncertain. The architecture has not changed, but the behaviour has been shaped. Most useful chat assistants live here.

A further step adds retrieval. Retrieval-augmented generation (RAG) has become the most common architectural intervention of the last several years8, and it deserves its prominence. By giving the model access to relevant documents at query time, RAG reduces the model's reliance on parametric memory and grounds outputs in retrievable sources. It is a real improvement, and it is one of the building blocks of what comes next.

But RAG, by itself, is not verification. It is retrieval. The model still produces the answer; the retrieved documents are context. Whether the answer actually reflects what those documents say is, in most RAG systems, not independently checked. The familiar failures – drift, overclaim, citation of one document for a claim that is actually only weakly supported by it – survive the addition of retrieval9, because nothing in the architecture is doing the comparison work. RAG retrieves. Verification evaluates. The two often appear together, but they are distinct moves.

Beyond RAG sit systems that begin to make the comparison itself a first-class step: evidence-grounded systems that check generated claims against retrieved material; systems that flag contradictions; systems that decline to answer when the evidence is thin. These are closer to what we mean by Verification AI. They have authority outside the model in a non-trivial sense.

At the far end of the spectrum are systems designed for the highest-stakes work — narrow scope, strict grounding, conservative refusal behaviour, full audit trails, human review built into the loop. They are less impressive in benchmarks. They are also the only systems that can stand behind their outputs in environments where someone will be asked to explain why.

The right place on this spectrum depends on what is being asked of the system. A creative-writing assistant does not need a contradiction-detection layer whereas a clinical summariser

does. The verification burden rises with the stakes, and a useful system pattern is one that lets the surrounding architecture be heavier where the consequences are heavier, and lighter where they are not.

Different systems beginning to rhyme

What is striking, watching the field over the last 2 years, is how independently different teams in different domains have begun to converge on similar components when the stakes get high enough.

The components have different names in different places. They are not yet a settled vocabulary. But they are the same components.

Grounded retrieval runs through all of them: claims tied to specific source material rather than to the model's parametric memory, with the link from claim to source preserved through the system.

Bounded scope – the discipline of saying out loud which questions a system will and will not attempt, and refusing the rest rather than answering them poorly.

Evidence comparison, a step distinct from generation, checks whether the generated draft actually reflects what the retrieved material says.

Contradiction detection surfaces, rather than smooths over, places where evidence disagrees with itself or with the draft.

And then human auditability: any consequential output will, eventually, be looked at by someone whose job is to decide whether to trust it. The system has to make its reasoning legible to that person.

These five recur, in different combinations and at different depths, across the systems that have been most successful at being trusted in high-stakes work. They are not a method. They are not a checklist. They are the shape that trustworthy systems seem to want to take.

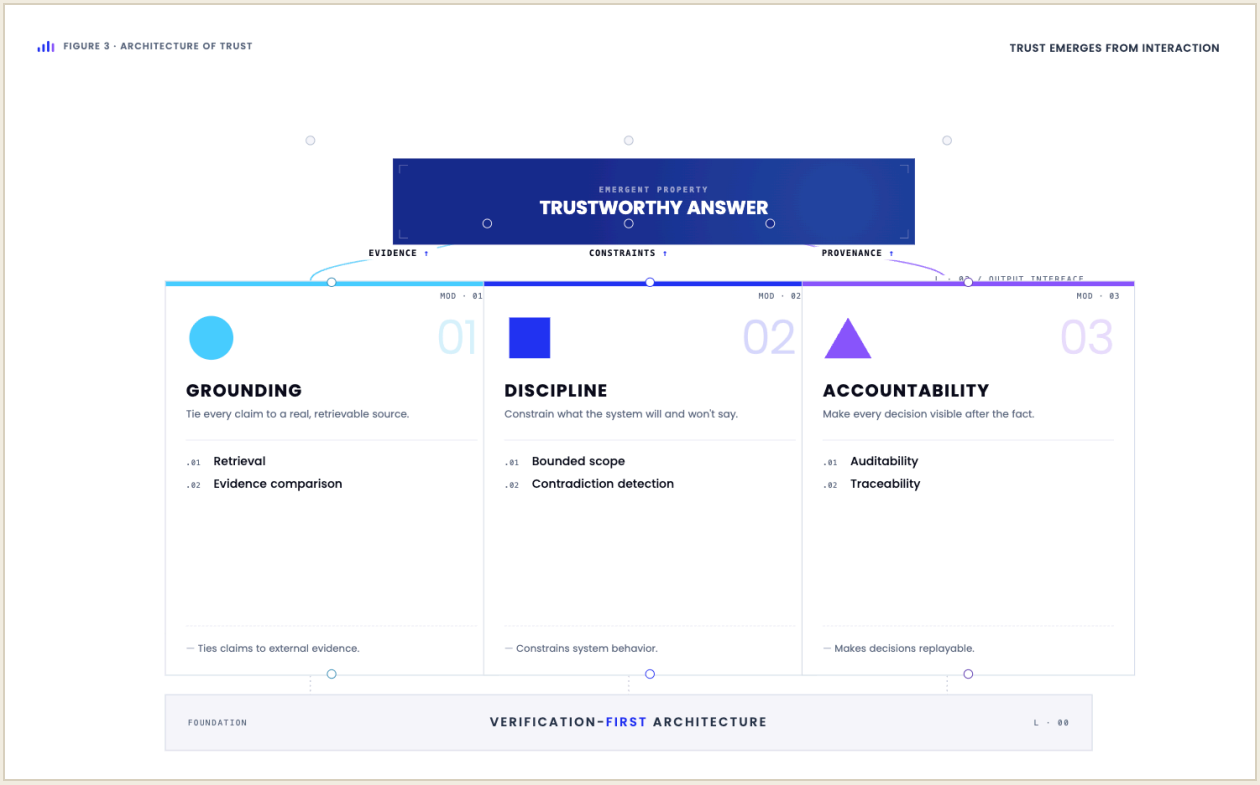

FIG. 3 Trustworthy answers emerge from the interaction of three architectural pillars: grounding (linking claims to evidence), discipline (constraining system behaviour), and accountability (making decisions auditable and traceable). Verification-first systems compose these mechanisms into a single trust architecture rather than relying on model capability alone.

The next section is about each of them in detail. The point here is the recurrence. When independent teams, working on different problems, with different incentives, end up building the same kinds of components, it suggests the components are responding to something real about the underlying task, not following a fashion.

A shift in posture

None of this is presented as a revolution. Verification AI is not an invention. It is a name for an emerging design pattern that becomes more necessary as AI is asked to do work where someone will be held responsible for the result.

What changes, with this pattern, is the aspiration. The old aspiration – implicit in a great deal of model-centric work – was to make the model always right. The new aspiration is more modest, and more useful: to make reliability legible. To build systems whose outputs come with the evidence behind them, the limits around them and the disagreements that complicate them, so that a competent reviewer can decide whether to trust each one. Legible reliability is, in this framing, the central deliverable of a trustworthy system. It is not as ambitious as omniscience, it is much more useful.

If this shift is real, the natural next question is what trustworthy verification systems have in common – what principles, repeated across different domains and different implementations, give Verification AI its shape? The question is no longer whether verification matters, but what verification, done well, consistently requires.

5

ch. 05

A Design Framework for Verification AI

— Trust is structural, not stylistic.

If trustworthy systems are beginning to converge on similar shapes, the next question is what those shapes are actually made of.

This section attempts to answer it. Not by proposing a framework – the field already has no shortage of frameworks – but by tracing what the rhyme is. The argument is empirical, in a soft sense. Across very different domains, very different teams, and very different model generations, the systems that earn trust in high-stakes work seem to develop similar architectural properties. Those properties are not rules. They are what trustworthy systems, given enough time and consequences, tend to grow.

There are five recurring components, named in the previous section: grounded retrieval, bounded scope, evidence comparison, contradiction detection and human auditability. They are useful as a vocabulary. But they fall into three deeper requirements that are easier to think about together: grounding (the way claims are tied to evidence), discipline (the way the system constrains itself) and accountability (the way reliability is made inspectable). These three are the actual shape. The five components are how that shape is realised.

What follows is an attempt to describe each requirement in turn – what it asks of the system, why it matters, what failures it prevents and where it is harder than it looks.

Grounding

The first thing trustworthy systems do is make sure their claims are connected to evidence, not just adjacent to it.

This is a stronger requirement than it sounds. Most systems built in the last few years touch evidence somewhere in their pipeline. Many retrieve documents. Many cite sources. A growing number do both. But there is a meaningful difference between a system that has access to evidence and a system whose outputs are accountable to it. Verification AI sits on the second side of that line.

Two components do most of the work. *Grounded retrieval* ties claims to specific source material, with the link from claim to source preserved through the system. *Evidence comparison* checks, in a step distinct from generation, whether the source actually supports the claim that has been built on top of it. Together they constitute grounding.

The first half, retrieval, has become standard practice. A system that does not consult source material at query time has, in most high-stakes settings, no defensible answer when asked where its claims come from. Parametric memory is opaque, untraceable and prone to slow drift across model generations. Retrieving the relevant material puts the evidence in the room, which is a precondition for almost everything that follows.

What is less appreciated is how much room is left for failure between “the evidence is in the room” and “the claim reflects the evidence.” This is the territory of the second component. A retrieved document is context, not control. The model still produces the answer, and the model is just as capable of overclaiming a citation, summarising it loosely, or quietly broadening its scope as it would be without the document present. Evidence comparison closes that gap by asking a separate question: does the source actually support what has been said?

It is a deceptively simple step, and it is where many of the most consequential failures live. In practice, these failures emerge not in retrieval – the right document was usually found – but in interpretation. The model read it, drafted from it, and moved one notch beyond what it said. The system did not notice, because nothing in the architecture was looking.

Comparison, done well, looks more like a structured check than a stylistic flourish. It asks specific questions: does the source contain this claim? Does it qualify it? Is the support direct or inferred? Are there other sources in scope that disagree? When the answers fail, the system has options – soften the claim, attach the qualifier, flag the discrepancy, refuse – and those options can be exercised without the user being asked to do the work themselves.

Grounding,

in this stronger sense, is what makes a system's answers checkable. It is also what makes them honest. A claim with no traceable basis is, in a high-stakes context, not really a claim. It is an assertion the user is asked to take on faith. Trustworthy systems do not ask their users to take things on faith. They make the path from evidence to claim visible enough that the question of trust can actually be asked.

Discipline

If grounding is about what trustworthy systems do, discipline is about what they refuse to do.

This is the requirement that surprises people most often, and the one that distinguishes mature high-stakes systems from impressive demos. The natural inclination of a generative system is to answer – to produce a confident, complete, fluent response to whatever it has been asked. Most of the time, that inclination is what makes the system useful. In high-stakes settings, the same inclination is the source of a disproportionate share of the trouble.

Two components carry the weight here: *bounded scope* and *contradiction detection*. They are different but related, and they point at the same underlying property – a system that knows the limits of its own competence and stays inside them.

Bounded scope is the discipline of saying out loud what the system will and will not attempt. A clinical system might commit to summarising and comparing licensed indications and decline to advise on individual cases. A legal system might commit to surfacing on-point precedent in a specific jurisdiction and decline to predict outcomes. A financial system might commit to flagging anomalies in filed disclosures and decline to recommend trades. The specific contents matter less than the act of drawing the boundary and enforcing it. Systems without that boundary will, eventually, be asked questions outside their competence and will, in the spirit of being helpful, attempt them. The attempt is where the failure begins.

A trustworthy system is often distinguished less by what it answers than by what it declines to answer. The refusal is not a limitation; it is a feature. It is the visible sign that the system has an honest view of its own capability and is willing to act on it. Refusal also reduces the surface area on which the surrounding verification machinery has to work – bounded scope makes everything else cheaper.

Contradiction detection is the other half of the same instinct. Real evidence in high-stakes domains rarely tells a single, clean story. Sources disagree. Guidelines update faster than databases. Two studies find opposite effects. A summary that flattens all of this into a single confident answer is, in most cases, less trustworthy than one that flags the disagreement and lets the reviewer see it.

This is where generative systems struggle most by default. Their training pulls them toward smoothing disagreement away, because a clean synthesis is rhetorically more satisfying than a hedged one. A trustworthy verification layer pushes the opposite way. It looks for places where the evidence is not unanimous, where the draft has resolved an ambiguity the source did not resolve, where two retrieved passages point in different directions. It surfaces these places rather

than hiding them. It treats unresolved disagreement as information, not as friction to be removed.

The reframe is small but important: trustworthy systems do not eliminate ambiguity, they make ambiguity visible. The reviewer is then in a position to make the actual judgment. The system has done its honest job by refusing to make the judgment for them.

Discipline, in both forms, runs against the grain of how generative systems naturally behave. That is precisely why it has to be architectural. A prompt asking a model to “refuse when uncertain” produces refusals roughly at the rate the model finds it natural to produce them. A bounded-scope layer that simply rejects out-of-domain queries produces them at the rate the domain requires. The same is true of contradiction surfacing. These behaviours have to be built into the system; the model alone will not get them right.

Accountability

Grounding makes claims checkable. Discipline keeps the system within its competence. Accountability is what makes both of these visible to the people who will rely on them.

This is the requirement that gets the most lip service and the least serious design attention. “Human in the loop” has become a default phrase for AI systems destined for regulated environments, but the framing is often weaker than it sounds. A human watching a system produce outputs is not the same as a human able to inspect why the system produced them. The two have very different implications for how the system is built.

The stronger framing is humans able to inspect the loop. The reviewer should be able to ask, of any consequential output, what evidence was retrieved, what was compared, what was flagged as contradictory, what the system declined and what level of confidence the underlying machinery had any right to assert. Those questions should have answers – specific, traceable answers, recorded at the moment the output was produced, not reconstructed afterward.

This is what we mean by legible reliability. It is not the same thing as the system being right. It is the system being checkable. A perfectly reliable oracle that produces answers without traceable bases is, from an institutional point of view, still not a trustworthy substitute for the work it replaces, because the institutions that depend on the work need to be able to certify it. The reviewers cannot review what they cannot trace; the clinicians, lawyers, analysts and editors who will stand behind the system’s outputs cannot stand behind anything they cannot inspect.

The architectural implication is that the verification work has to leave a trail. The retrieval that brought back evidence; the comparison that checked the draft; the contradiction layer that surfaced disagreements; the scope boundary that refused certain questions – each of these has to produce a record that survives long enough to be useful. The record is part of the deliverable, not exhaust from it.

In practice, an audit trail for a single high-stakes output might include the documents retrieved, the passages supporting each generated claim, the comparison decisions about whether each claim is well supported, the contradictions surfaced and how they were handled, the scope checks that determined the question was in bounds, and the confidence signals exposed to the reviewer. None of that is hard to log. What is hard is committing, at design time, to producing outputs that justify the logging – to treating every consequential answer as a candidate for inspection rather than a finished product.

The shift in framing matters. The weaker version of trust is “trust the model” – believe the output because the system that produced it has been built carefully. The stronger version is

“trust the process by which the answer was produced” – and that requires the process to be visible. A trustworthy system is not one that asks to be believed. It is one that makes belief inspectable.

This is also the requirement that connects Verification AI most directly to the institutions it will eventually serve. Medicine, law, science, finance, regulated communications – these are domains that have built, over decades, elaborate machinery for making expert work checkable. They will not, in our experience, treat AI systems as legitimate substitutes for that work until the systems themselves produce something inspectable enough to slot into the same machinery. Accountability is the design property that makes that possible.

A shape, not a checklist

Taken together, grounding, discipline and accountability describe a posture more than a method. They are what a system looks like when it has been built with the assumption that someone will be asked to defend its outputs. None of them are flashy. None of them improve a benchmark score in the way a capability upgrade would. But they are what trustworthy systems consistently grow toward, and the recurrence is the point.

It is worth acknowledging what this costs. Verification introduces friction. Retrieval steps, comparison passes, refusal logic, contradiction surfacing, audit logging – each of these adds latency, complexity and engineering effort relative to a system that just lets the model answer. A consumer assistant for casual writing does not need any of it; a clinical decision-support system arguably needs all of it. The verification burden is meant to rise with the stakes, not to be imposed uniformly. Proportionality is part of the design, not an afterthought.

None of this is novel in its components. Grounding has been a concern of retrieval systems for years. Discipline echoes long-standing software practice around scope and safety. Accountability is what every regulated profession has had to develop in some form. What is new is the recognition that all three need to coexist inside the same system when that system is generative, and that the absence of any one of them leaves a recognisable failure mode in its place. The three requirements are not a discovery. They are an inheritance – adapted from older fields that have already learned to do this kind of work.

What the three share is that they all move work that generative systems do not naturally do – verifying, refusing, surfacing disagreement, leaving a trail – out of the hopeful realm of “the model should do this if we ask nicely” and into the structural realm of the surrounding system. That move is the architectural content of the generation-to-verification shift. The five components named in the previous section are how it is implemented. The three requirements named here are what those components are for.

Healthcare did not invent these requirements. It simply exposed them earlier than most domains have been forced to confront them, because the cost of getting them wrong arrives sooner there than almost anywhere else. It is therefore the natural place to examine how Verification AI behaves under pressure.

6

ch. 06

Healthcare as the Proving Ground

— Where stakes are high, the surrounding system carries the weight.

Healthcare's role in the history of trustworthy AI is unusually informative, but not for the reasons that get most often cited.

The familiar argument is that healthcare matters because the stakes are high – that mistakes harm patients, that the consequences of failure are severe, that this elevates healthcare above other domains. That argument is true as far as it goes but it is also incomplete. Plenty of domains have high stakes. The reason healthcare is worth attending to closely is not that its stakes are uniquely severe. It is that healthcare combines four properties – evidence richness, ambiguity richness, consequence richness and institutional density – in a way that makes weak trust architectures fail visibly, and fail fast.

This combination is the proving ground. Trustworthy systems that work in healthcare tend to work elsewhere; systems that fail in healthcare tend to reveal more clearly what they were missing. The principles from the previous section (grounding, discipline, accountability) are not specifically clinical principles. But they show up earlier in clinical settings than in most others, because clinical settings stress all three at once.

What follows is an attempt to explain why, in terms of the four properties named above. None of this is an argument that healthcare is exceptional in some final sense. It is an argument that healthcare is early – that the architectural pressures the rest of high-stakes AI will eventually feel are already visible there.

Evidence-rich

Healthcare is one of the few domains where almost every meaningful claim has, at least in principle, a traceable evidentiary lineage.

Drug indications trace to regulatory submissions, which trace to clinical trials, which trace to specific protocols and outcomes. Guideline recommendations trace to evidence syntheses, which trace to bodies of underlying studies. Even routine clinical knowledge – dosing, contraindications, monitoring intervals – sits on top of citable bodies of literature. The infrastructure is not perfect, and the literature itself is uneven, but the expectation of traceability is built into the culture.10 A clinician asking “where did this claim come from?” is asking a routine question, not an unusual one.

That expectation has two consequences for AI systems. The first is straightforward: there is something to ground against. Almost any consequential clinical claim a system might generate can, in principle, be checked against a specific source – a label, a guideline, a regulatory document, a trial report. The grounding work that Verification AI requires is not theoretical here; the evidence base exists, is mostly accessible and is structured enough to be retrievable.

The second consequence is sharper. Healthcare is unusually intolerant of unsupported confidence. A clinical reader will not, by default, accept a fluent paragraph at face value the way a casual reader might. The professional habit is to ask for the basis. A system that cannot produce one – or that produces a citation that fails on inspection – loses credibility quickly and does not get it back. Confident prose without traceable evidence is not just an imperfection in this environment; it is a tell.

This rewards architectures that take grounding seriously. It also punishes shortcuts. A generation-first system that hallucinates citations will be caught faster in healthcare than almost anywhere else, because the reviewers know what intact citations look like.

Ambiguity-rich

If evidence richness rewards grounding, ambiguity richness rewards discipline.

The thing that surprises people coming into clinical AI from other software backgrounds is how rarely the evidence settles a question cleanly. Clinical questions live in a fog of contested studies, contradictory trial results, evolving guidelines, subgroup effects that complicate population-level recommendations, off-label realities that depart from on-label evidence and time-dependent shifts as new data arrive. The right answer is often qualified, conditional or contested. Sometimes the honest answer is that the question itself is malformed for the patient at hand.

Generative systems are poorly equipped for this kind of terrain by default. They are trained to produce answers. A clinical question with no clean answer is not a natural prompt for them; they will produce something that sounds like an answer, often by quietly resolving ambiguities the underlying evidence does not resolve. The result is fluent, confident, and – to a reviewer who knows the evidence – visibly wrong in a way that is hard to forgive.

What clinical work rewards instead is what could be called faithful uncertainty: the willingness to qualify, to attribute, to flag disagreement, to decline. A summary that says “two trials disagree on this endpoint, and the most recent guidelines weight one over the other for the following reason” is more useful, in a clinical setting, than a summary that picks a side without showing its work. The former is something a reviewer can act on; the latter is something a reviewer has to second-guess.

This is why the discipline requirements of the previous section – bounded scope and contradiction detection – are so important. Bounded scope is what lets a system refuse a question that the evidence does not actually support an answer to. Contradiction detection is what lets the system, when the evidence is mixed, present the mixture honestly rather than collapse it. Both behaviours feel native in healthcare because the alternative – confident synthesis over messy ground – fails so visibly.

In medicine, confidence is not always competence. Trustworthy systems in this environment tend to be less assertive than they could be, on purpose. It is one of the ways the architecture earns its credibility.

Consequence-rich

Stakes in healthcare are usually discussed in terms of catastrophic outcomes, but most of the relevant trust pressure does not come from catastrophes. It comes from the accumulation of small failures that, over time, erode confidence in the system.

A confidently misquoted contraindication. A summary that subtly shifts the indication. A response that elides a known interaction. None of these is, in isolation, a catastrophe. Each is the kind of error a competent clinician will spot, sometimes immediately. But the cumulative effect of being asked to spot such errors, repeatedly, is the slow withdrawal of trust.11 The system becomes something the user has to check rather than something the user can rely on, and the value of the system collapses long before any individual error reaches the threshold of harm.

This is one of the less-appreciated dynamics of clinical AI deployment. Teams sometimes assume that, because their systems do not produce headline-worthy failures, the deployment is going well. The signal they are missing is the quieter one: that reviewers have started reading every output as if it were suspect, that workflows have re-routed to avoid relying on the system for anything that matters, that the system has been demoted from tool to suggestion box. Once a clinical environment has been taught to distrust a system, it is very difficult to teach it to trust the system again.

This dynamic puts a particular kind of pressure on the design. The relevant target is not “high enough average accuracy that catastrophes are rare.” It is “high enough reliability per output that the system can survive being checked routinely.” Those are different targets. The first tolerates a long tail of small errors; the second does not.

Healthcare makes reliability visible because the consequences of weak reliability arrive quickly. Not as harms – those would be rarer – but as withdrawal. Architectural choices that produce a stream of confident, almost-right outputs do not get to the harm stage. They get caught at the trust stage, and the system stops being used. This is one of the ways healthcare punishes architectures that look adequate in benchmarks.

Institutional density

The fourth property is the one that most directly shapes what trustworthy systems in healthcare have to look like.

A clinical AI system is not deployed into the relationship between a model and a user. It is deployed into a dense fabric of institutions that have spent decades developing infrastructure for managing human fallibility. Regulators evaluate evidence and approve indications. Medical boards certify professional competence. Peer-reviewed publication validates findings. Clinical guidelines synthesise evidence into actionable recommendations.12 Audit trails record decisions and their bases. Malpractice law assigns responsibility when things go wrong. None of this infrastructure was built for AI, but all of it now applies to AI by extension.

The implication is that healthcare AI inherits, by default, expectations that AI systems in other domains have not yet been confronted with at the same intensity. The system's outputs will be reviewed by professionals trained to look for specific kinds of error. They will be checked against established evidence bases by people who know those evidence bases well. They will, if the system is consequential enough, be subject to formal regulatory oversight. They will, in the event of a bad outcome, attract questions about traceability and accountability that the system has to be able to answer.

This is not a barrier to AI in healthcare. It is the design environment. A system built without these expectations in mind will not survive contact with the institutions it is meant to serve. A system built with them in mind has to develop, by necessity, the properties the previous section described – claims tied to evidence, scope bounded honestly, refusals on uncertain ground, audit trails that survive inspection. The institutions, in effect, do a lot of the architectural work for the AI team. They make it impossible to ship a system that lacks accountability, because nothing without accountability will be accepted.

The objection sometimes raised at this point is that healthcare's institutional density makes it unrepresentative – that the verification pressure described here is an artefact of unusually heavy regulation, and that other domains face nothing comparable. There is some truth to this. Healthcare has more institutional density than most environments AI is currently entering. But the relevant question is not whether other domains have comparable density today; it is whether the underlying pressures – evidence expectations, ambiguity, consequence, accountability – exist elsewhere, and whether the institutional structures around them will harden as AI moves in. The pattern of this whitepaper suggests they will. Healthcare is where the structures are already mature. It is also where AI systems are first finding out what mature structures actually demand.

Not exceptional, just early

What the four properties produce together is something more useful than a portrait of clinical AI. They produce a clear early view of what trustworthy AI will look like, more broadly, as it moves into work that matters.

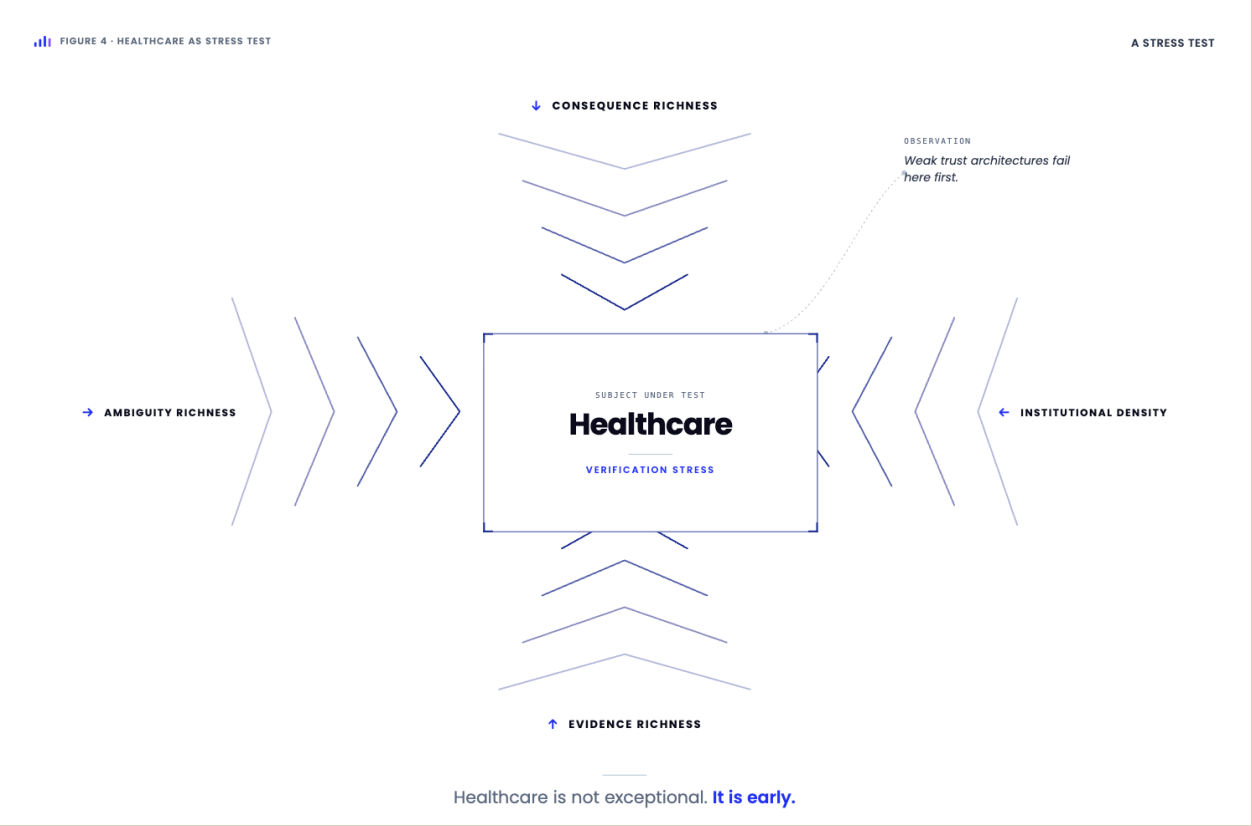

FIG. 4 Healthcare functions as a stress test for trustworthy AI systems. High consequence richness, ambiguity richness, institutional density, and evidence expectations create unusually strong verification pressure. Weak trust architectures fail here first, making healthcare an early indicator of broader shifts toward verification-first AI.

The same pressures that healthcare brings to bear now are already beginning to appear elsewhere. Legal work has its own version: evidence in the form of statutes and precedent, ambiguity in the form of jurisdictional and interpretive complexity, consequence in the form of clients and courts, institutional structure in the form of bar associations and rules of professional conduct. Financial analysis has another: regulatory filings as evidence, ambiguity in the interpretation of disclosures, consequence in the form of market impact and compliance liability, institutional structure in audit and reporting frameworks. Scientific publishing, enterprise knowledge work, policy analysis – each of these has the same signature, at different intensities, with infrastructure that is more or less mature.

The

pattern in each case is the same. Where the four properties are strong, generation-first systems struggle and verification-first systems take hold. Where the properties are weak, lighter architectures suffice. Healthcare is informative not because it is unique but because all four are unusually strong at once, which compresses the architectural learning curve. Mistakes that would take years to surface in less constrained domains surface in months in clinical deployment. Verification architectures that would emerge slowly in finance or law are emerging faster in healthcare because the environment is less forgiving of their absence.

Healthcare does not change the problem. It accelerates it. The architectures that prove their value there are the architectures that, in time, the rest of high-stakes AI will need.

What that means for those other domains is the subject of the next section.

7

ch. 07

Beyond Healthcare — Verification in Other High-Stakes Domains

— Authority is earned, not asserted.

If the previous section is right that healthcare is early rather than exceptional, the natural next move is to look at the domains where the same pressures are starting to appear. Healthcare's four properties – evidence richness, ambiguity richness, consequence richness and institutional density – are not unique to clinical work. They sit, in different combinations and at different intensities, in every domain where AI is being asked to produce claims that someone else will rely on.

What follows is a wider view of the same architectural pressures, organised less around individual industries than around the recurring pattern. The same three architectural requirements (grounding, discipline, accountability) keep showing up. The form they take is different in each domain. The shape underneath is not.

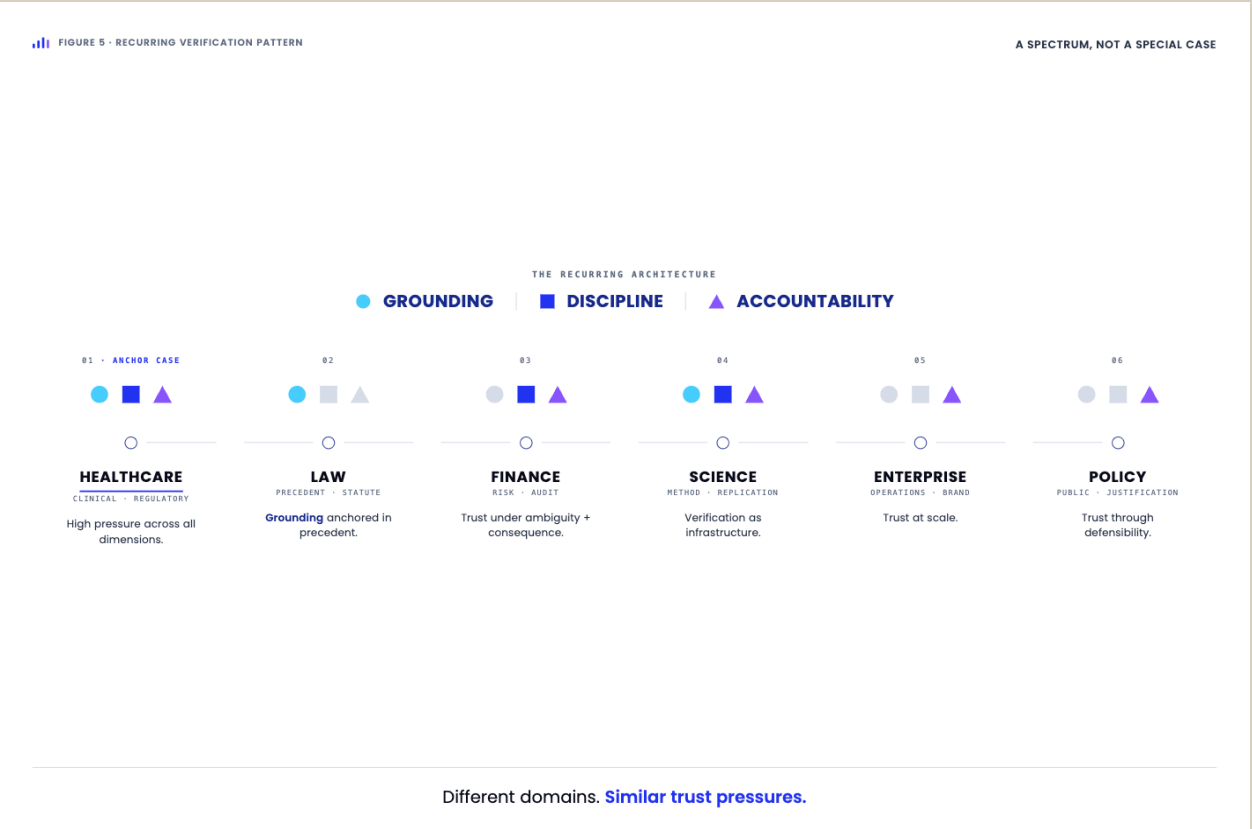

FIG. 5 Verification-first architectures recur across high-stakes domains, though the relative emphasis differs. Healthcare serves as the anchor case because trust pressures converge early and intensely, but law, finance, science, enterprise systems, and policy exhibit similar patterns of grounding, discipline, and accountability under different conditions.

Law: where authority already runs on evidence

Law was the first domain in which generative systems produced an unambiguous public failure that the broader world could understand. The fabricated case is by now a well-known genre13: a confidently written brief citing precedents that, on inspection, do not exist. The story was instructive less because it was scandalous than because it was structurally inevitable. Generative systems optimise for the shape of citations, not for their existence. Law happens to be a domain where the shape and the existence are unusually separable, and where the failure is unusually easy to detect.

What makes law informative for the wider argument is not the citations themselves. It is that legal work has always run on grounding. Statutes, precedents, regulations, jurisdictions – every meaningful legal claim is, in principle, traceable to a source that another lawyer can pull off the shelf and read. The expectation of inspectable reasoning is built into the profession. A lawyer who cannot show where a claim comes from is a lawyer with a problem, regardless of whether the claim turns out to be right.

This makes law a domain where Verification AI is almost defined by what generation alone cannot do. A system that drafts arguments persuasively but cannot tie them to specific authorities is producing the shape of legal reasoning without its substance. A system that drafts persuasively and grounds correctly is doing the thing the profession actually needs. The difference between those two systems is architectural.

Finance: where almost-right is expensive

Finance is, on the surface, less obviously evidence-bound than law. The documents are different – earnings releases, regulatory filings, analyst reports, internal models – but the underlying pattern is similar. A claim about a company's risk profile, the implications of a covenant, the meaning of a footnote in a disclosure: all of these have, in principle, a checkable basis. Whether the system that produced the claim actually checked against it is a separate question.

Where finance sharpens the pattern is on ambiguity. Real financial interpretation is rarely a clean exercise. Disclosures are written to satisfy legal requirements as much as to inform; numbers carry context that is easy to miss; the same metric can mean different things across industries, periods, and accounting regimes. A system that produces a confident reading of an ambiguous filing is producing exactly the kind of fluent, almost-right output that an earlier section described – and in finance, almost-right can be expensive. An overstated risk, a missed caveat, a confident extrapolation from weak support: these are the failures that, repeated at scale, move capital in directions the underlying evidence does not support.

Like law, finance has spent decades building infrastructure for making analysis defensible – audit trails, methodology disclosures, model documentation, regulatory review. AI systems entering this space are entering it on those terms. The systems that survive will be the ones whose claims are traceable, whose limits are stated, and whose disagreements with the data are surfaced rather than smoothed.

Science: where verification already runs

Science is the domain where the architectural argument of this paper most directly meets an existing tradition.

Verification is not a new idea in science. Peer review, citation, reproducibility, methods transparency, evidence hierarchies – the entire infrastructure of scientific work is, in effect, a verification system that humans built around their own fallibility long before AI was part of the conversation. The cultural expectation that a claim must be defensible, traceable, and challengeable by competent peers is the closest analogue, anywhere, to what Verification AI is trying to produce in software.

This creates a productive tension. Generative systems are unusually good at the surface of scientific writing: fluent prose, conventional structure, plausible argumentation, well-formatted references. They are unusually poor at the underlying property the writing is supposed to signal, which is that the claims it contains have been verified. A scientific paper produced by a system that drafts confidently and cites loosely is the exact analogue of the legal brief with fabricated cases, but the failure mode is harder to spot, because scientific writing is full of dense citations that even careful reviewers do not usually check one by one.14

The risk is not faster writing. The risk is the slow degradation of the trust infrastructure that scientific work depends on – fabricated references in published literature, citation graphs polluted by claims that were never actually supported by the cited work, interpretation inflation that survives review because the prose around it sounds right. Science gets the architectural problem more clearly than perhaps any other field. Its credibility depends less on what can be written than on what can be defended.

Enterprise knowledge work: the quiet frontier

Enterprise knowledge work is the quietest of these domains, and one of the most consequential.

An internal assistant that summarises company policy, fields procedural questions, drafts technical documentation, or explains a compliance requirement is not, in any single instance, doing high-stakes work in the dramatic sense. But it is doing work that thousands of employees will rely on without independent verification, because reducing verification effort is often the point of using the system in the first place. The failure mode is not catastrophe; it is the slow contamination of internal knowledge with confidently wrong answers.

The dynamic resembles the trust-erosion pattern described in clinical AI. An enterprise assistant that produces a stream of almost-right answers does not announce its failures. It quietly miscalibrates the workforce. Decisions get made on the basis of misremembered policy. Procedures get followed in subtly wrong ways. New employees learn the company's rules from a system that does not always know them correctly. The cost shows up not as a single visible failure but as a gradual decline in the reliability of what people inside the organisation believe about the organisation.

The architectural response is the same as elsewhere. The internal assistant has to be grounded in real source material – the actual policy document, the actual procedure – rather than drawing from parametric memory of similar things it has seen. It has to know which questions sit outside its competence and refuse them. It has to leave a record that allows a compliance team to ask, later, where a particular answer came from. None of this is exotic. It is just verification, applied to the kind of work that does not usually get described that way.

Policy and governance: the slowest-moving domain

Policy and governance are the slowest-moving domain on this list, and the one where the verification pressure is least visible today and most consequential over time.

The work itself – regulatory interpretation, policy briefing, evidence synthesis, comparative analysis – looks tractable for generative systems. Most of the inputs are documents. Most of the outputs are documents. The leap from “model that reads policy” to “model that writes policy briefings” is not technically large. What is large is the gap between producing a briefing that reads well and producing one that can survive being relied on by a decision-maker who will, eventually, be asked how their decision was reached.

Policy decisions accrete consequences in a way that is unusually hard to undo. A regulation enacted, a programme funded, a precedent set – each carries forward through institutions for years. The defensibility of how the underlying analysis was produced is, accordingly, more important than in domains where outputs are more easily revised. The more consequential the decision, the harder it becomes to separate trust in the answer from trust in how the answer was produced.

This is the domain in which the institutional structures around AI are still least mature. They will harden, and probably faster than people expect, because the alternative – confident, fluent, well-formatted policy analysis whose evidentiary basis cannot be reconstructed – is not a tenable input to legitimate governance.

What rhymes across domains

Across these domains, the variation is real and the pattern is consistent. Law's verification problem is most immediately about grounding; finance's is most acutely about discipline under ambiguity; science's is about preserving an existing trust infrastructure; enterprise work's is about preventing slow trust erosion; policy's is about defensibility under accumulated consequence. The mix is different in each case. The underlying requirements are the same.

What makes the recurrence interesting is that it is not driven by anyone deciding that these domains should look architecturally similar. It is driven by the underlying properties – evidence, ambiguity, consequence, accountability – which exist in each of these fields independently of whether AI is involved. As AI enters, it inherits the demands those properties produce. The systems that adapt to those demands begin to develop similar shapes, for the same reasons trustworthy human-built systems in these domains have always developed similar shapes.

The point is not that every domain becomes healthcare. Most do not. The verification burden is proportional to the underlying properties; a creative-writing assistant remains a creative-writing assistant, and lighter architectures continue to suffice where the stakes are lower. What is changing is the upper end of the spectrum – the kinds of systems we are willing to call trustworthy in environments where someone will be asked to defend the output. At that end, the architectural pattern is recognisable across domains in a way that was not obvious a few years ago.

The future of trustworthy AI may look less like universal intelligence than like domain-specific systems that know how to justify themselves. If this pattern is real, the next question is what it demands of the people building these systems, the institutions deploying them, and the research agenda shaping them. That is where the paper turns next.

8

ch. 08

Implications for Designing Trustworthy AI

— Capability is a property of the model. Trust is a property of the system.

If the previous sections are right that trustworthy high-stakes AI is converging on verification-first architectures, several things follow that are not always foregrounded in current AI discussions. They are not recommendations. They are consequences of the argument, visible once the argument is taken seriously.

This section traces five of them. Each is an evolution rather than a rupture – the kind of shift that becomes obvious in retrospect and contested in the moment. Taken together, they amount to a different picture of where the meaningful work on trustworthy AI is likely to happen over the next several years.

The centre of gravity shifts from models to systems

The most consequential of these is that the centre of gravity in trustworthy AI begins to move outward from the model.

For most of the modern AI era, the model has been the thing. Capability, reliability and safety have all been attributed to it. The surrounding architecture is treated as a thin shell – prompting, deployment plumbing, perhaps some retrieval – through which the model's properties are delivered to users. The mental model is model-first; the model is what is being trusted.

The argument of this paper suggests that, in high-stakes work, this picture inverts. The unit of trust is no longer the model. It is the system the model sits inside, including the components that ground, constrain, audit and check what the model produces. The model is the most powerful component in that system, but it is one component among several, and the trustworthiness of the system is not reducible to the trustworthiness of any one part.

The practical consequence is that the engineering question shifts. Where it used to be "which model should we use?" – a substitution question, answerable by benchmarks and pricing – it becomes "what verification structures surround the model?" – a design question, answerable only by reasoning about the specific work the system is doing and the trust it needs to earn.

This is not anti-model progress. Better models make the surrounding system cheaper and easier to build, because the components can be lighter when the model is more reliable. But better models do not, on their own, build the system. The system has to be designed.

Evaluation becomes harder, and more realistic

Evaluation gets harder before it gets better.

Most of the evaluation infrastructure that has grown up around large language models is built for a generation-first world. Benchmarks measure what models can do on well-defined tasks, scored against reference answers, often at scale. These measurements are useful, and they have driven real progress. They are also poorly aligned with what trustworthy deployment actually requires.

The properties that make a system trustworthy in high-stakes work – appropriate refusal, faithful uncertainty, contradiction surfacing, evidence-grounded restraint, auditable reasoning – do not benchmark cleanly.15 They show up on the long tail rather than the average. They are most important in exactly the cases where there is no clean reference answer to score against. A system that achieves a strong benchmark score by being confidently correct most of the time can still fail in deployment, because the trust-eroding behaviours of the previous sections cluster in the cases the benchmark does not capture.

This is not a problem evaluation can solve by being more careful in the same direction. It is a problem that requires different kinds of measurement: evaluations that test refusal as a positive behaviour, that score systems on how well they surface disagreement rather than how cleanly they resolve it, that reward grounding quality and audit completeness over fluency. Some of this work is starting, but it is early and not yet standardised. The gap between what we can measure and what we need to measure is the gap that determines, in practice, whether a system will survive deployment in environments where someone is checking.

Reliability in deployment is not the same thing as benchmark performance. The two correlate, but the correlation weakens exactly where the stakes rise.

High-stakes AI becomes narrower as models become broader

Another implication runs against the grain of how AI progress is usually narrated.

The dominant story is one of generalisation. Models get bigger and broader, capable of more things across more domains, with fewer task-specific accommodations. That story has been broadly correct for the underlying systems. It is likely to remain correct for some time. But it sits in tension with what trustworthy deployment in high-stakes domains actually requires.

The systems that earn trust in clinical work, in legal practice, in financial analysis, in regulated communications are not the most general systems. They are the most disciplined ones. They are the systems that have explicitly committed to a bounded scope – a particular kind of summary, a particular kind of comparison, a particular kind of refusal – and that have built the verification machinery around that scope to make it stand up to inspection. Discipline is what makes them defensible. Generality, in this context, is what makes them suspect.

The implication is a likely divergence in how AI capability and AI deployment evolve. Capability will continue to broaden, because that is how the underlying technology improves. Deployment in serious domains will, for the same reasons, narrow – into systems with clearer commitments about what they do and do not attempt. The trustworthy systems of the next few years may look less impressive than the underlying models that power them, precisely because their value comes from what they refuse rather than from what they attempt.

Trustworthy systems may become narrower as models become broader. That is not a regression. It is what specialisation looks like in a field whose underlying capability has finally become rich enough that not every system has to attempt everything.

Institutions begin shaping architecture

Institutions, often treated as passive adopters of AI, become active architects of it.

The conventional view of how AI enters regulated domains is that the technology arrives, the institutions adapt, and the regulators eventually catch up. The argument of this whitepaper suggests something closer to the reverse. AI systems that enter healthcare, law, finance, science, and regulated enterprise work inherit the trust infrastructure those institutions have already built. They cannot ignore it. The institutions, through their evidence expectations, audit norms, accountability structures, and professional cultures, end up exerting architectural pressure on the systems that operate inside them.

This pressure does not feel architectural at first. It feels like friction – regulatory questions, review boards, professional scepticism. But the systems that survive the friction are the ones that have, by necessity, developed the properties the friction was demanding all along: traceable claims, scoped competence, surfaced disagreement, inspectable reasoning. What initially looks like institutional friction often turns out to be architectural pressure. The friction is the architecture, expressed in institutional rather than software terms.

This reframes who is doing the design work. Institutions may, in effect, be the hidden architects of trustworthy AI – shaping it not through design documents but through what they enforce, accept, and reject. A clinical guideline committee that requires citations is doing verification design. A regulator that requires methodology disclosure is doing verification design. A bar association that defines what a lawyer is allowed to delegate is doing verification design. None of them would describe themselves that way. All of them are.

Recognising this changes how the work proceeds. AI teams that try to ship trustworthy systems without engaging the institutional context spend a long time learning, by failure, what the institutions could have told them earlier.

Safety shifts from behaviour to structure

The deepest of these, and the one most likely to be contested, is structural.

A large fraction of current AI safety work focuses on the behaviour of models – how they respond to certain prompts, how they refuse harmful requests, how they handle adversarial inputs, how their values align with intended objectives.16 This work is important. It is not the work this paper is about. The work this paper is about is structural: what happens around the model, what gets logged, what gets checked, what gets refused, what gets surfaced, what gets traced. The two kinds of safety address different parts of the same underlying problem.

Behavioural safety is about making the model do the right thing. Structural safety asks a different question: what system properties make failures survivable, visible and correctable when the model does the wrong thing? Both are necessary, and neither is sufficient on its own. A model that behaves well in 99.9% of cases will still produce failures at scale; the question is whether the surrounding system makes those failures into events that can be detected and addressed, or whether they propagate silently. A system that catches and corrects failures effectively but is built around a model with poor instincts is paying a high cost for that catching. The two have to coexist.

What this paper adds to that conversation is the observation that structural safety has been underweighted. The attention that goes into alignment – the kind of work that aims to make models more reliable from the inside – has not been matched by attention to the architectures that make reliability legible from the outside. Both directions are required if trustworthy AI is to be more than an aspiration.

Reliability may ultimately depend less on perfect behaviour than on imperfect systems that fail honestly. That is the structural intuition, and it is the one this paper has been arguing should sit alongside the behavioural one.

A question, not a programme

These five implications do not amount to a programme. They are consequences of taking the verification argument seriously – what becomes visible once trust is treated as a property of systems rather than of models alone.

What ties them together is a quiet shift in what the central question of AI engineering is. For most of the past decade the question has been how intelligent models can become. That question has been productive, and it has not stopped being productive. But it has been joined, in the domains that matter most, by a different question – one that the rest of the field will increasingly have to engage with.

The question is no longer how intelligent models can become. It is what kinds of systems deserve trust. The two questions are related, but they are not the same, and they have different design implications. The first is mostly about the model. The second is mostly about everything around it.

The first question built the current era of AI. The second may shape what trustworthy AI becomes.

What that looks like in practice, and where it leaves the broader argument, are the subjects of the remaining sections.

9

ch. 09

From Principle to Practice

— In a trustworthy system, the model is the easy part.

The argument of this whitepaper has not been formed in the abstract. It has been shaped, gradually, by direct work building verification-first systems for medical writers, patient communication, and pharmaceutical compliance review (developed under PharmaTools.AI). What follows is not a survey but a brief account of three applied systems – different in purpose, related in architecture – that have served as practical tests of the design ideas described in earlier sections.

RefCheckr is an evidence-verification system for medical writers. A user uploads a draft alongside the source documents the draft draws on, and the system checks each substantive claim against those documents. It surfaces supporting passages, flags claims for which no supporting passage can be located, and identifies apparent contradictions between the draft and its sources. Every claim it reports comes attached to a specific passage – or, conspicuously, to the absence of one – so the output is not an opinion about the draft but a traceable record a human reviewer can interrogate. The intent is not to replace the writer's or reviewer's judgement but to do the thing the human reviewer in practice rarely does at scale: actually compare the prose to the underlying documents, line by line. Many claim–evidence mismatches surface before a draft reaches human review, and the contradictions that do surface tend to be the kind that fluency would otherwise have hidden.

Patiently AI is a patient-communication tool built around a deliberate constraint: translation, not interpretation. Given clinical material – a discharge note, a result, a piece of trial information – it produces clearer, plainer-language versions for patient-facing use. It does not diagnose or recommend treatment. It does not add clinical content the source material did not already contain. The design choice underneath all of this is that the system is permitted to clarify what is already said, but not to add what is not. Much of the verification work is, in this sense, negative: containing the model's natural tendency to elaborate beyond its evidence.

MedCheckr is a compliance-oriented system that supports pharmaceutical regulatory review by checking promotional and informational content against the relevant code clauses – the kind of judgement an internal medical or regulatory reviewer performs before a piece of material is approved. The system is not asked to decide whether copy is compliant. It is asked to surface, for each potentially relevant clause, the specific passage of the content the clause applies to and the apparent fit or tension between the two. The compliance judgement remains with the human reviewer. What changes is what is visible to them, and what is traceable in the record.

The three systems share an architectural posture rather than a common feature set. They are not designed to be impressive in demonstration. They are designed to produce outputs that can be checked, bounded, traced, compared, and – when wrong — corrected. The model in each case is the easiest component to upgrade. The verification structure around it is where most of the engineering, and most of the judgement, actually lives.

of this is offered as proof of the broader argument. It is offered as evidence that the broader argument is at least implementable: that verification-first design is not only a theoretical position but a practical posture already being explored, with the usual unevenness, in regulated medical communication systems. The interesting question is not whether such systems can be built. It is what becomes possible – and what becomes harder – when they are.

10

ch. 10

Conclusion — The Missing Layer

— Intelligence may generate the answer. Trust is what the surrounding system earns.

This whitepaper began with a distinction. LLMs are becoming remarkably capable. Whether they are becoming trustworthy is a different question, and not one that progress on the first reliably answers.

The intervening sections have argued that the difference matters more than the field has fully reckoned with. Capability and trustworthiness improve at different rates and for different reasons. They are sometimes assumed to converge – that a sufficiently intelligent system will, by virtue of its intelligence, also be a trustworthy one. The argument of this paper is that they do not converge automatically. Trustworthiness is built on top of capability rather than delivered by it, and the work of building it is architectural – done by people thinking, often quietly, about the structures that surround the model rather than the model itself.

That argument has consequences. In the domains where AI is being asked to inform decisions that someone will be held responsible for – clinical, legal, financial, scientific, regulated enterprise, policy – the systems that earn trust are converging on a particular shape. They ground their claims in evidence. They limit the questions they will attempt. They surface disagreement rather than smoothing it. They leave records that allow their reasoning to be inspected. They treat the model less as an oracle than as one component inside a structure that does the verification work the model alone cannot.

This is what we have called Verification AI. It is not a product, a method, or a framework. It is the shape trustworthy systems seem to want to take when the stakes are high enough to make their absence visible.

A pattern, not a trend

What makes the argument of this paper feel – to the people who have lived inside it – like a description rather than a proposal is how consistently it shows up without being prescribed.

Teams in healthcare, working on different problems with different timelines and different incentives, end up building similar things. So do teams in law. So do teams in finance, in regulated science, in enterprise compliance. The components are sometimes named differently. The order of construction varies. But the underlying shape – grounding, discipline, accountability – appears with a regularity that is harder to attribute to fashion than to gravity.

The reason, the paper has argued, is that trust in high-stakes work has structure. The same four properties – evidence, ambiguity, consequence, accountability – keep producing the same architectural demands. The systems that meet those demands keep developing the same kinds of components. The result is not a trend but a convergence: independent teams, under independent pressures, arriving at recognisably similar designs because the underlying problem rewards them.

Verification AI is the name for what those designs have in common. It is less an invention than a recognition: a way of describing what systems grow into when trust matters enough to drive their architecture. The next several years will, almost certainly, refine the vocabulary. The shape underneath has already begun to settle.

Proportional, not universal

None of this is an argument that every AI system has to be built this way.

Generation-first systems remain enormously useful. Creative work, brainstorming, drafting, low-stakes assistance, exploratory thinking – these are domains where the model's natural inclination to produce a confident answer is the source of the value, not the source of the risk. The verification burden in these settings is light because the trust burden is light. A creative-writing assistant does not need a contradiction-detection layer, and demanding one would be an obstacle to what makes the assistant useful in the first place.

What changes with the stakes is the requirement, not the technology. Higher stakes demand heavier verification; lower stakes do not. The right architecture is the one proportionate to what the system is being asked to do. The argument of this paper is not that verification is everywhere. It is that where decisions matter, trust increasingly has to be earned structurally rather than asserted by the model.

The missing layer

For most of the past decade, the working assumption in AI has been that the next generation of trustworthy AI would arrive with the next generation of intelligent AI – that capability gains would, eventually, deliver reliability gains as a side effect. The history of the field has, in some ways, been a long demonstration of how powerful this approach can be. Each successive model has done things its predecessor could not. Each has moved problems that looked architecturally hard into the category of problems that scale can solve.

But not every problem is in that category. The trust problem in high-stakes work has turned out to be the kind of problem that does not collapse cleanly under capability. It is not solved by making the model smarter, or more fluent, or more confident. It is solved, where it is solved at all, by building structures around the model that decide what its outputs deserve. Those structures are the work this paper has been describing.

The missing layer in trustworthy AI was never intelligence. It was architecture – the design of the system around the model, the discipline of grounding what it produces, the institutional and engineering scaffolding that makes its reliability legible to the people who depend on it. None of this is mysterious work. It is just work that has been overshadowed, until recently, by the work on the model itself.

Capability will continue advancing. The harder, slower work is deciding which of its outputs deserve trust. The future of trustworthy AI will be shaped, at least in part, by the people willing to do that work, and by the institutions willing to insist on it.

Intelligence may generate the answer. Trust is what the surrounding system earns.

References

1. Mata v. Avianca, Inc., 678 F. Supp. 3d 443 (S.D.N.Y. 2023), https://law.justia.com/cases/federal/district-courts/new-york/nysdce/1:2022cv01461/575368/54/.

2. Anthropic, Claude Opus 4 & Claude Sonnet 4 System Card (May 2025), https://www.anthropic.com/claude-4-system-card.

3. E. M. Bender, T. Gebru, A. McMillan-Major and S. Shmitchell, “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?”, Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT ’21), 610–623, https://dl.acm.org/doi/10.1145/3442188.3445922.

4. M. Sharma et al., “Towards Understanding Sycophancy in Language Models”, Anthropic, arXiv:2310.13548 (2023), https://arxiv.org/abs/2310.13548.

5. K. Goddard, A. Roudsari and J. C. Wyatt, “Automation bias: a systematic review of frequency, effect mediators, and mitigators”, Journal of the American Medical Informatics Association 19 (2012), 121–127.

6. N. F. Liu et al., “Lost in the Middle: How Language Models Use Long Contexts”, Transactions of the Association for Computational Linguistics 12 (2024), 157–173.

7. S. Kadavath et al., “Language Models (Mostly) Know What They Know”, Anthropic, arXiv:2207.05221 (2022), https://arxiv.org/abs/2207.05221.

8. P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”, Advances in Neural Information Processing Systems 33 (NeurIPS 2020), https://arxiv.org/abs/2005.11401.

9. V. Magesh, F. Surani, M. Dahl, M. Suzgun, C. D. Manning and D. E. Ho, “Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools”, Stanford RegLab / HAI Working Paper (May 2024), https://law.stanford.edu/publications/hallucination-free-assessing-the-reliability-of-leading-ai-legal-research-tools/.

10. G. H. Guyatt et al., “GRADE: an emerging consensus on rating quality of evidence and strength of recommendations”, BMJ 336 (2008), 924–926.

11. J. S. Ancker et al., “Effects of workload, work complexity, and repeated alerts on alert fatigue in a clinical decision support system”, BMC Medical Informatics and Decision Making 17, 36 (2017).

12. Institute of Medicine, Clinical Practice Guidelines We Can Trust (Washington, D.C.: National Academies Press, 2011), https://nap.nationalacademies.org/catalog/13058/clinical-practice-guidelines-we-can-trust.

13. D. Charlotin, AI Hallucination Cases (database), https://www.damiencharlotin.com/hallucinations/.

14. W. H. Walters and E. I. Wilder, “Fabrication and errors in the bibliographic citations generated by ChatGPT”, Scientific Reports 13, 14045 (2023).

15. P. Liang et al., “Holistic Evaluation of Language Models (HELM)”, Transactions on Machine Learning

Research (2023).

16. Anthropic, Responsible Scaling Policy, https://www.anthropic.com/responsible-scaling-policy.