KINDYNOS ADVISORY
EVENT INTELLIGENCE

Model Risk Governance When the Model Does Not Repeat Itself

The supervisors have now conceded the mismatch. What follows is the harder question.

Santa Federico · Kindynos Advisory · July 2026 (revised)

Model Risk Governance When the Model Does Not Repeat Itself

The supervisors have now conceded the mismatch. What follows is the harder question.

Santa Federico · Kindynos Advisory · July 2026 — revised

> **Note on this revision.** I first wrote this paper as an argument that supervisory model-risk procedure and generative AI were mismatched, and that the object of validation had to move. In April 2026 the US banking agencies rescinded SR 11-7 and OCC Bulletin 2011-12 and issued replacement interagency guidance — which **explicitly excludes generative and agentic AI from its scope**, describing the technology as novel and rapidly evolving. The argument in this paper is no longer a prediction. The mismatch has been conceded by the supervisors themselves, and institutions deploying these systems now do so in an acknowledged gap. That changes the paper's purpose: it is no longer making a case that something is wrong, but proposing what should fill the space the guidance has deliberately left open. I have revised it accordingly.

Summary

Supervisory guidance on model risk management was written for models that return the same answer twice. Generative models do not. This is not a gap in coverage that a new procedure will close; it is a mismatch between a control framework and the object it is asked to control.

That mismatch is now official. In April 2026 the Federal Reserve, OCC, and FDIC jointly issued revised interagency model risk management guidance, rescinding SR 11-7 and OCC Bulletin 2011-12. The revised guidance preserves the core principles — validation, monitoring, governance, effective challenge — while adopting a more risk-based and materiality-sensitive posture. And it **excludes generative and agentic AI from its formal scope**, on the stated ground that these technologies are novel and rapidly evolving. Institutions are told to apply existing risk-management principles in the meantime, and further guidance is expected.

I want to be precise about what that means, because it is easy to read as either reassurance or alarm and it is neither. It is not a deregulation. It is a supervisory body declining to write procedure it does not yet believe it can write well — which is, in my view, more honest than the alternative and considerably more useful than a premature rulebook. But it leaves institutions deploying the fastest-growing category of model in an acknowledged gap, holding the principles and none of the procedures.

The principles survive contact with generative AI almost entirely. The **procedures** do not. Every practical technique the old guidance relied on — replication as the test of

soundness, validation as a point-in-time exercise, benchmarking against a champion model, documentation of a specification an independent party can re-implement — assumes determinism somewhere in its foundations. That is precisely why the new guidance declined to extend to systems that have none.

This paper argues for a shift in what is validated. **Stop trying to validate the model. Validate the system around it.** A model risk committee cannot sensibly approve a distribution over outputs. It can approve a system that constrains which outputs reach a decision, records what produced each one, and refuses when it cannot substantiate an answer.

Six requirements follow, and they are buildable today:

1. **Provenance as a first-class output** — every result carries what produced it
2. **Typed refusal** — the system declines rather than guessing, by name
3. **Scoped reliance** — the output states how far it may be relied upon
4. **Absence is never a number** — a missing value never appears as zero
5. **Continuous evaluation** — validation as a standing control, not an event
6. **Identity and pinning** — a change in what produced an answer is a visible change

I close with the failure mode that concerns me most, which is not hallucination. It is the passing test suite that measures the wrong thing.

A note on what this paper is for. With the supervisory gap now explicit, an institution that waits for guidance will wait while deploying. The six requirements below are what I would put in place in the interim, and they are constructed so that whatever guidance eventually arrives will find an institution with evidence rather than intentions.

1. Why the existing framework strains

The supervisory definition of a model — a quantitative method applying statistical, economic, financial, or mathematical theories to process input data into quantitative estimates — is one a generative language model fits without difficulty. The revised 2026 guidance retains that definitional approach, and most traditional machine-learning models in banking clearly fall within it. The strain was never definitional.

It appears in the machinery. And it is worth walking through the machinery specifically, because understanding *which* procedures fail explains why the agencies drew the scope line where they did.

Replication. The guidance leans heavily on the idea that an independent party can reproduce a model's output from its specification and inputs. This is the backbone of effective challenge. With a temperature above zero, the same prompt against the same weights yields different text. Even at temperature zero, reproducibility is fragile across hardware, batch composition, and inference-stack versions. The validator cannot re-derive the number, so the central evidentiary act of validation is unavailable in its usual form.

Conceptual soundness. A validator is expected to assess whether the theory and design are appropriate for the intended use. For a logistic regression this is tractable. For a hundred-billion-parameter transformer, the honest answer is that no one can assess conceptual soundness at the level of the mechanism. Interpretability research is real and advancing, but it does not yet support a validation opinion on why a particular output was produced.

Benchmarking. The standard technique is comparison against an alternative — a champion, a challenger, a simpler model. When the output is free text or a multi-step agentic trajectory rather than a number, there is often no natural alternative estimator and no metric on which to compare.

Ongoing monitoring. Monitoring assumes a stable object being monitored. Vendor models are updated, sometimes with little notice; system prompts change; retrieval corpora change; tool definitions change. The model in production next quarter may not be the model that was validated, and the institution may not be the party that changed it.

Documentation. The guidance asks for documentation detailed enough that a knowledgeable third party can understand the model. For a vendor foundation model, the institution does not have the training data, the weights, or the training procedure, and will not get them.

None of this means generative AI cannot be governed. It means the object of governance has to move.

2. The principles that survive

It is worth being clear about what does not change, because there is a temptation in institutions to treat novelty as a licence for exemption.

The following survive intact:

- **Model risk is real and must be managed.** The possibility of adverse consequences from decisions based on incorrect or misused output is, if anything, greater with generative systems, because output fluency is decoupled from output reliability in a way it never was with a regression.
- **Effective challenge remains the core control.** Critical review by informed, competent, and *empowered* parties. What changes is what they examine.
- **Use is part of the model.** Supervisory doctrine has always held that a model used outside its intended purpose is a source of risk, and the 2026 revision's materiality-sensitive posture sharpens the point rather than softening it. With generative systems this becomes the dominant consideration, because the same model serves an unbounded range of uses and the boundary of intended purpose is set by the deploying institution, not by the model.
- **Ownership and accountability.** Someone owns the model, someone independently validates it, and someone is accountable for the framework.
- **Proportionality.** Governance intensity scales with materiality and consequence.

The framework's authors were not naïve about non-linear or opaque methods. What they could not anticipate was a model whose output is a *sample* rather than a *value*.

The 2026 revision confirms this reading. It preserved validation, monitoring, governance, and effective challenge — the principles — while declining to extend its procedural apparatus to generative and agentic systems. Read carefully, that is the same distinction this paper draws: the principles hold, the procedures do not, and the honest response is to say so rather than to pretend a replication test can be performed on a sampler.

It also means the instruction institutions have actually been given is to *apply the principles without the procedures*. That is a demanding instruction and it has almost no published detail behind it. The rest of this paper is my attempt at that detail.

3. The shift: validate the system, not the sample

Here is the reframing that makes the problem tractable.

A generative model does not produce an answer. It produces a **distribution over answers**, from which one is sampled. Asking a validation committee to approve a distribution is asking the wrong question, and the discomfort practitioners feel in these meetings is the discomfort of being asked something unanswerable.

But an institution does not deploy a model. It deploys a **system**: retrieval, prompt construction, tool access, the model, output parsing, constraint checking, escalation rules, human review, and the decision the output feeds. The model is one component. Every other component is deterministic, inspectable, and entirely amenable to conventional validation.

So move the object of governance:

> **Do not ask whether the model is correct. Ask whether the system reliably prevents an unsubstantiated output from reaching a decision.**

That question can be answered. It can be tested, evidenced, and re-tested. And it is the question the institution actually cares about, because no one has ever been harmed by a model's internal distribution — they are harmed by a decision made on an output that should not have been trusted.

The rest of this paper is what such a system must do.

4. Six requirements

I have spent the last eighteen months building an execution engine for consequential decisions, and these six requirements are not aspirational. They are implemented, and I will illustrate each with what the implementation actually does.

4.1 Provenance as a first-class output

Every result must carry, in the result itself rather than in a log to be correlated later: which model, which **version**, which inputs, which state it read, and what operation produced the value.

In practice this means a result is not a number. It is a record:

```
Δ = -5.664602505722238 [VALUED; derivation DIFFERENCE; model
pricing::bsm_quantlib@1.43; pre 5.669114454465682 @ spot=127.62; post
0.004511948743444246 @ spot=85.0]
```

The number cannot be separated from its derivation, because they are the same object. An examiner asking "where did this come from" is answered by the artifact, not by a reconstruction.

For a generative component the same discipline applies: model identity and version, the exact prompt, the retrieved context with document versions, the tools called and their responses, and the sampling parameters. If reproduction is impossible, **recording** must be complete. This is the substitution that replaces replication: you cannot re-derive it, so you must be able to reconstruct exactly what happened.

4.2 Typed refusal

The system must be able to decline, and the decline must be **typed** — a specific machine-readable reason, not a generic failure.

```
event market_crash → the_call/OPTION_NPV = UNVALUED (BACKEND_VERSION_MISMATCH)
[execution: REFUSED; reliance: PROHIBITED] detail: requested=1.42:available=1.43
state: v0 → v0  [not committed; published 0] exit 3
```

Three things are worth noting. The system refused rather than using the version it had. The reason is specific enough to act on. And **nothing was committed** — a refusal leaves no residue in state.

The generative analogue: when a model cannot ground an answer in retrieved evidence, when a tool call fails, when output fails a constraint check, or when confidence falls below threshold, the system must produce a typed refusal that propagates. It must not produce a plausible answer with a quiet caveat.

This is the requirement institutions find hardest, because refusals are visible and wrong answers often are not. A system that refuses ten per cent of the time looks worse in a demonstration and is worth enormously more in production.

4.3 Scoped reliance

An output should state how far it may be relied upon, and that stamp must be derived rather than asserted.

Three levels are sufficient in practice: **reliable within a stated scope**, **limited** (something in the computation did not fully apply), and **prohibited** (do not use this for anything).

The critical property is that reliance must degrade **automatically** when the system's own conditions are not met. In my implementation, an allocation whose declared scope could not be fully applied is capped at `LIMITED` and can no longer report itself as

reliable — not because a human noticed, but because the code path that would have stamped it reliable is unreachable when the scope was not applied.

Reliance that a human sets is documentation. Reliance the system computes is a control.

4.4 Absence is never a number

> **A missing value must never be rendered as zero, and an absence must never be rendered as a number.**

This is the oldest lesson in institutional risk and the one most frequently re-learned. A position that could not be priced is not a position worth nothing. A counterparty exposure that failed to load is not zero exposure. A control that was not tested is not a control that passed.

Generative systems make this worse, because they are fluent. A model asked for a figure it does not have will often supply a plausible one. A retrieval system that returns nothing will often be summarised as "no issues identified." Both are absences rendered as values, and both are indistinguishable from real answers by the time they reach a decision-maker.

The system must distinguish, in its type system rather than by convention:

- a genuine zero — **valued**, and equal to zero
- a value that could not be computed — **unvalued**, with a reason
- a value not applicable in this context — **not applicable**, with a reason

An honest zero and a missing value must never be the same object.

4.5 Continuous evaluation

Point-in-time validation was defensible when models were stable. It is not defensible when the vendor may update the model, the retrieval corpus changes weekly, and behaviour drifts without any change the institution initiated.

Validation becomes a **standing control**: an evaluation suite that runs continuously, on every change, with results retained as a time series. It must include known-answer cases, adversarial cases, refusal cases (does the system still decline what it should decline?), and regression cases derived from every defect ever found.

The organisational consequence is significant. Model validation groups are staffed and budgeted as project-based functions performing discrete reviews. Continuous evaluation is an engineering function with an operational footprint. Institutions that do not make this shift will govern their generative systems on paper while the systems change underneath them.

4.6 Identity and pinning

A change in what produced an answer must be a **visible** change.

This means version pinning, exactly: a component may declare the version it requires, and a version that cannot be honoured causes a refusal rather than a silent substitution. It also means the configuration should have an identity — a hash over the model version, prompts, tool definitions, retrieval configuration, and constraints —

such that two systems differing in any material respect are recognisably different systems.

Without this, the sentence "the model was validated" has no referent. With it, an institution can state precisely which system was validated and detect immediately when the deployed system is no longer that one.

5. What an examiner can be given

The practical question in any supervisory conversation is what actually goes in the binder. Under this framework it is not a model specification and a replication test. It is:

1. **A system identity** — exactly what was validated: model versions, prompts, tools, retrieval configuration, constraints, and the hash over them.
2. **The evaluation suite and its history** — what is tested, why those cases, results over time, and every regression case with the defect that produced it.
3. **The refusal inventory** — what the system declines to answer and how each refusal is triggered. This is the most informative single document and almost nobody produces it.
4. **The reliance policy** — what each stamp means and which decisions may be made at each level.
5. **Provenance samples** — actual production records showing complete lineage, including refusals.
6. **The escalation and human review record** — what reached a human, what they did, and how often they overrode.
7. **Change control** — every change to any element of the system identity, with the evaluation results before and after.

An institution that can produce those seven artifacts is in a materially stronger supervisory position than one holding a conventional validation report on a model it cannot replicate.

6. The failure mode I would examine first

Not hallucination. Hallucination is well known, widely discussed, and increasingly mitigated by grounding and constraint checking. The failure mode that concerns me is quieter.

A test suite can be entirely green while measuring the wrong invariant.

An example from my own system, which I record because it is more instructive than a hypothetical.

For several development stages, a portfolio calculation reported an allocation of **+216,502.33** to a stakeholder holding a position that had **lost 16.92**. Wrong sign. Wrong by four orders of magnitude. And stamped with the system's own highest reliance level.

The full test suite passed throughout — every stage, every build.

The cause was mundane: the allocation logic summed every effect in the calculation rather than the effects of the asset the allocation actually named. But the reason it survived was not the defect. It was the shape of the tests:

> **Every regression check constrained a magnitude. Not one constrained an attribution.**

The totals were always right. The portfolio total never moved by a cent. So the regression floor held perfectly while the split beneath it was wrong — for months, in every affected calculation, under a reliable stamp.

The general form of this is the question worth putting to any validation programme, generative or otherwise:

> **When your test suite passes, what class of error is it constructed to be unable to see?**

For generative systems in financial institutions, I would expect the answer to be attribution and justification rather than magnitude. Evaluation suites overwhelmingly measure whether the answer was right. Far fewer measure whether the *reason given* was the reason the system actually used, whether the citation supports the claim, or whether the confidence expressed tracks the evidence available. A system that is right for the wrong reason will eventually be wrong for the same reason, and nothing in a conventional accuracy metric will give warning.

A second lesson from the same audit, which generalises further than I expected. We found six fields the system accepted from its users and then wholly or partly ignored. The framing that came out of it:

> **A field the language accepts and the engine ignores is a lie the system lets the user tell without being contradicted.**

Applied to generative deployment: every configuration option, guardrail toggle, and policy setting that is accepted but not enforced is exactly this. It is worth auditing which of your controls are actually read.

7. The gap is not only American

It would be a mistake to read the US position as an outlier, and equally a mistake to assume another jurisdiction has solved what the agencies deferred.

The EU AI Act reaches financial services directly and by name: creditworthiness assessment and life and health insurance pricing are classified as high-risk, which places most retail-facing bank and insurer AI inside the heaviest obligations — risk management, data governance, technical documentation, logging, human oversight, and post-market monitoring. But its timetable has moved. Under the 2026 Digital Omnibus agreement, high-risk obligations for standalone Annex III systems were deferred from August 2026 to **2 December 2027**, and for AI embedded in regulated products under Annex I to **2 August 2028**. Certain Article 50 transparency obligations for systems already on the market moved to December 2026. An institution that built

its programme against the original calendar now has more time than it planned for, and should resist the temptation to use it as slack.

Nor do the voluntary frameworks close the gap. The NIST AI Risk Management Framework, ISO/IEC 42001, and the OECD principles are the instruments most institutions are layering underneath their regulatory obligations, and they are genuinely useful — but none of the three was designed for agentic systems. At the time of writing, Singapore's 2026 framework is the only governance document I am aware of that addresses autonomous agents directly.

The picture, then, is consistent across jurisdictions rather than peculiar to one: **the supervisory apparatus for generative and agentic systems in financial services does not yet exist anywhere, and the bodies responsible have been candid about it.** That candour is welcome. It also means the interim is longer and wider than a single agency's deferral would suggest, and that an institution's own controls are, for now, the whole of the control environment.

8. What to do on Monday

For an institution deploying generative AI into consequential decisions:

1. **Inventory by consequence, not by technology.** Which decisions can this system affect, and what happens if it is wrong? Governance intensity follows from that, not from whether something is labelled AI.
2. **Move the object of validation** from the model to the system, and say so explicitly in the framework document. Half the difficulty in these programmes is that people are being asked to validate something that cannot be validated, and nobody has said so out loud.
3. **Require provenance and typed refusal in procurement.** If a vendor system cannot tell you why it refused or what produced an answer, that is a governance defect, not a feature request.
4. **Build the refusal inventory first.** It is the fastest way to discover that a system has no refusals — which is the finding, and a serious one.
5. **Make validation continuous** and staff it as engineering.
6. **Audit your evaluation suite for what it cannot see.** Ask specifically whether anything tests attribution and justification rather than only outcomes.
7. **Establish system identity and change control** before scale, not after.

None of this requires new supervisory guidance — which is fortunate, because for generative and agentic systems there currently is none, and the agencies have said so plainly. This is an application of the surviving principles to an object the retired procedures were not built for, in the interval before whatever replaces them arrives.

There is a second reason to act rather than wait. When guidance does come, it will be written by people looking at what institutions actually did in the gap. An institution that can show a refusal inventory, a provenance record, a continuous evaluation history, and a system identity will be describing a practice. An institution that waited

will be describing an intention. The first is a much better position from which to receive a rule, and a much better position from which to influence one.

9. A closing note on where the difficulty really sits

The hardest part of this is not technical. Every requirement in §4 is buildable with current tools, and I have built them.

The difficulty is that a well-governed generative system is **visibly worse** than a poorly governed one in every setting where these systems are evaluated. It refuses more. It answers more slowly. It qualifies its outputs. It says "I cannot substantiate this" where a competitor says something fluent and confident. In a procurement bake-off, the honest system loses.

That asymmetry is exactly the one institutional risk management has always existed to correct, and the argument for correcting it is the same one it has always been. The cost of a confident wrong answer is not paid at the demonstration. It is paid later, by someone who was not in the room, and usually at a scale that makes the efficiency gain look trivial in retrospect.

I have spent four decades on the receiving end of that trade, and eighteen months building a machine whose central commitment is to refuse rather than guess. I am fairly confident the trade has not changed.

What has changed is the alibi. Until this year an institution could reasonably say that supervisory expectations for these systems were unsettled and it was waiting for direction. The direction has now arrived, and it says: the principles apply, we are not yet writing you procedures, use your judgement. That is an unusual amount of latitude, and latitude is only comfortable if you are confident in what you would do with it. The institutions that come out of this period well will be the ones that treated an acknowledged gap as an obligation rather than a reprieve.

A note on currency: this paper was revised in July 2026. The regulatory position it describes changed twice in the preceding six months — the US interagency guidance in April and the EU timetable in May — and should be verified before reliance. Nothing here is legal or compliance advice.

Santa Federico is the founder of Kindynos Advisory. He was Global Head of Market Risk at Salomon Smith Barney and Citigroup, and Chief Risk Officer at Perry Capital, Ally Financial, and Silver Ridge Asset Management, with regulatory engagement including Federal Reserve examinations, CCAR, and Basel implementation. He holds an A.B. in Physics from Princeton and a degree from École Centrale de Paris, and began his career at Bell Laboratories. He has been writing production software since 1977. This paper draws on EOS, an execution engine for consequential decisions built over the past eighteen months.

Comments and disagreement are welcome — the arguments here are working positions, not settled ones.