De-Identified Case Study — Governed AI in a Federal Healthcare Facility

An illustrative case study Author: Brad M. Lindsey — independent engineer · Master Electrician · Master HVAC Technician **Version:** 1.0 — 2026-06-02 **Companion to:** the policy white paper, one-pager, and pilot proposal.

> **Calibration banner (binding).** The workflows below are **illustrative demonstrations of governed-AI behavior**, drawn from a working diagnostic prototype. They are **not measured field results**, and no clinical or operational outcome is claimed. The facility is generic — "a federal healthcare facility" — with no name, no system identifiers, and no facility-specific details. The author writes as an independent practitioner; no employer, contract, or specific facility is implicated. Each workflow is paired with the **metrics a pilot would capture**, which are targets, not results.

Why these three workflows

A federal facilities engineer's real question about AI near critical systems is not "can it answer?" — it is "what does it do when it shouldn't answer?" These three workflows are chosen because they show the governance behavior, not just the capability:

1. A **governed answer** — the system gives a safe, specific first move *with its constraints attached*.
2. A **refusal to guess** — the system fails loud when the evidence isn't there, instead of producing a confident wrong answer.
3. A **contradiction caught** — the system flags when new evidence conflicts with what it was told before, instead of silently drifting.

The setting is a federal healthcare facility with NEC 517 essential electrical systems and an aging maintenance workforce — the kind of place where a confident wrong answer is a safety event, and where a retiring expert's troubleshooting instinct is exactly what is being lost.

Workflow 1 — A governed first move (the happy path)

Situation. An operator reports: a 20A breaker on a 240V panel trips immediately on reset; no smoke, no burning smell.

What an ungoverned assistant might do. Offer a plausible-sounding cause and a fix, with the same confidence whether or not it has the evidence to support it.

What the governed system does. It returns a structured, bounded response:

- **First Move:** *Isolate the load-side conductors and re-attempt the reset* — the only safe first action that doesn't energize an unknown circuit.

- **Do Not:** *Do not repeatedly reset the breaker under fault conditions* — an explicit prohibition, because repeated reset risks equipment damage and arc flash.
- **Next Required Fact:** *Does the breaker hold with all load disconnected?* — the single measurement that tells you whether the fault is upstream or downstream.
- **Provenance:** every part of that answer is traceable to the captured expert reasoning behind it, and the full exchange is recorded in an append-only audit trail.

Why it matters. The value is not that it named a likely cause. It is that it led with the *safe* action, attached an explicit prohibition, and told the operator the next fact to go get — the exact shape of how a veteran tradesperson thinks, captured and handed forward. The "Do Not" is the part an ungoverned model omits and an expert never would.

Pilot metrics this workflow would capture (targets, not results): mean time-to-diagnose for this fault class, expert vs. assisted-novice; whether the assisted operator took the safe first action; training-time-to-competency on this system.

Workflow 2 — A refusal to guess (the fail-loud path)

Situation. An operator submits a problem with key evidence missing — not enough information to responsibly produce a diagnosis.

What an ungoverned assistant might do. Fill the gap. Produce a confident answer anyway, because producing fluent text is what it does by default. This is the single most dangerous failure mode near a critical system.

What the governed system does. It refuses to proceed and says why. It returns an explicit failure — a `FAIL` state with a *"required to proceed"* clause naming the specific evidence it needs — rather than guessing. No diagnosis is offered until the missing fact is supplied.

Why it matters. This is the trust question, answered in the behavior. A system that knows the boundary of its own evidence — and stops at it, loudly — is one a safety officer can reason about. An operator learns quickly that when this system *does* answer, it has the basis to. The refusal is a feature, and it is the feature ungoverned AI conspicuously lacks.

Pilot metrics this workflow would capture (targets, not results): governance-integrity rate — how often the system correctly flags low-confidence or unverifiable cases instead of asserting; operator trust calibration over the pilot period.

Workflow 3 — A contradiction caught (provenance in action)

Situation. Earlier in a troubleshooting sequence, the evidence record holds "load-side conductors clean." New evidence arrives: a phase-to-phase short on a load-side conductor. The two cannot both be true.

What an ungoverned assistant might do. Quietly incorporate the new input and move on, losing the record that its picture of the system just changed — the slow drift that makes an AI's later conclusions untraceable.

What the governed system does. It detects the conflict against its prior asserted evidence and raises it explicitly — a contradiction flag naming the earlier fact and the new one — rather than silently overwriting. The audit trail preserves both states and the moment the picture changed.

Why it matters. This is provenance doing its job. On a live system, the history of *why* a conclusion was reached — and when the evidence shifted — is the difference between a diagnosis you can defend and one you can only hope is right. Catching the contradiction is also how the system stays honest over a long troubleshooting session, the same way an append-only ledger keeps a record you cannot quietly rewrite.

Pilot metrics this workflow would capture (targets, not results): rate of correctly flagged evidence conflicts; auditability — can a third party reconstruct *why* each conclusion was reached from the trail alone.

What the three workflows demonstrate together

Behavior	The failure mode it prevents	The pilot metric it maps to
Governed first move (with Do-Not + next fact)	A plausible answer that skips the safe action	Time-to-diagnose; safe-action rate
Refusal to guess (fail-loud)	A confident wrong answer when evidence is missing	Governance-integrity rate
Contradiction caught	Silent drift; untraceable conclusions	Conflict-flag rate; auditability

None of this is a claim that the prototype has been validated in the field. It is a claim about *how a governed system behaves* — and that behavior is exactly what the pilot in the companion proposal would measure under real conditions, with success criteria set in advance and results published either way.

The point of the case study is narrow and important: the hard part of putting AI near critical infrastructure was never capability. It was trust. These three workflows show what trust looks like when it is built into the system's behavior instead of promised in a disclaimer.

Illustrative workflows are drawn from a working diagnostic prototype in the author's portfolio. Underlying methodology: the Lindsey Provenance Discipline (`lindsey-provenance`, MIT) and its companion papers on the public record. Facility details de-identified throughout per the package guardrails.

Author's disclosure. This case study and the prototype it describes are LLM-collaborative work produced under that same provenance discipline — written to its own standard.

Calibration attestation. Every claim here is held to the proof-state on record as of June 2, 2026. The workflows are illustrative governed-AI behaviors, not measured field results; every pilot metric named is a target to be measured. No fabricated figures or unvalidated outcome claims appear.