# From SBOM to Decision BOM

## HOW INVENTORIES CREATE FALSE CONFIDENCE IN AI DECISIONS

## AUTHOR'S NOTE

An earlier essay, *Evidence Debt*, argued that the need to produce evidence grows as an AI-supported function gains authority. Evidence Debt is the gap between the evidence a function needs to support its decisions and the evidence it can actually produce.

This essay starts where that one left off. If you record everything involved in a decision, the model, the prompts, the data, the policies, the tools, and the human who approved it, have you proved the decision was sound?

The answer is no.

A complete record can show that a decision happened and what contributed to it. It still does not tell you whether that evidence was enough for the function, the consequence, or the authority involved. That distinction is easy to miss because a complete inventory feels reassuring.

The Decision Bill of Materials is a genuine advance, and I want it built. It gives organizations a much better record of AI-supported decisions. This essay focuses on the distance between a record that is complete and a decision that is justified. Software supply chains encountered a similar problem when they learned that knowing a component is present does not tell you whether it creates a real risk in that system.

These are my own views, not my employer's, and not a forecast from any company.

The framework is mine. I used an LLM to pressure-test the argument and help produce this designed PDF.

By Stephen Alexander

---

Companion work · cirrusly.me · Evidence Debt

## THE ARGUMENT IN BRIEF

An inventory can create confidence because it makes previously opaque systems easier to inspect. Software Bills of Materials did that for software supply chains. They also exposed a second problem: knowing that a component is present does not tell you whether it creates a real risk in a particular system. VEX was developed to answer that separate question.

Decision Bills of Materials are beginning to play a similar role for AI-supported decisions. They can record the model, data, policies, tools, and human approval involved. That is a real advance. A complete record can show how a decision was made. It does not, by itself, establish that the evidence was sufficient to justify the decision.

This essay extends the *Evidence Debt* argument by examining that distinction. It explains how to evaluate a decision record, why completeness and sufficiency are different, and how different evidence gaps require different responses.

Contents

PART I

# The Limits of a Complete Record

For most of computing's history, a shipped application was an opaque object: you could run it, but you could not see what was inside it. Then, in December 2021, a vulnerability in a small logging library called Log4j turned out to sit inside more than ten million projects, and organizations discovered they could not answer the question "are we running it?". The regulated response was the Software Bill of Materials, defined in Executive Order 14028 as a formal record of the components and supply chain relationships used in building software, and later mandated in the EU's Cyber Resilience Act. An SBOM turns the opaque artifact into a checkable inventory: seven minimum fields, standard formats, machine-readable.

It worked, and it taught a second lesson almost immediately. A complete SBOM tells you a vulnerable component is *present*. Whether *you* are exposed depends on your configuration, your usage, whether the vulnerable path is even reachable, none of which an inventory contains. The gap was real enough that a second artifact had to exist: VEX, the Vulnerability Exploitability eXchange. VEX reaches a finding. This product is exploitable, or it is not, stated as one of four plain statuses. It stops there. What to do about that finding, whether to patch tonight, accept the risk, or wait, belongs to the mission or business owner. An inventory is necessary and insufficient at the same time, and its value comes from tying its contents to a specific obligation or risk model. The rest of this essay applies that lesson to AI decision records.

The relevant artifact is no longer just a record of software components, but a record of the decision itself. As AI systems began making and executing consequential choices, two engineering communities converged, in 2025 and 2026, on the same move: apply the bill-of-materials idea to the decision itself. The result is the Decision Bill of Materials, a per-decision record of what produced an outcome: the model and prompt versions, the data consulted, the tools invoked,

the policies in force, the human actions taken. When a customer asks why a system denied them, when a regulator asks what the model saw, when an incident review asks who authorized the action, the DBOM is the artifact you open. It turns “the system decided” into a set of facts a person can inspect afterward. Two independent efforts arrived at that idea from opposite directions: one builds the record as the decision occurs; the other reconstructs it afterward from operational telemetry.

An academic construction out of Saarland and Dresden builds the record cryptographically: each DBOM is signed and generated inside a trusted execution environment, so a decision traces end-to-end to responsible stakeholders, on hardware that exists today. Microsoft ships the pragmatic mirror image: its agent-governance toolkit *reconstructs* the record afterward from the telemetry a system already emits, and scores each reconstruction from 0.0 to 1.0 for completeness against a five-field schema. One route is rigorous and forward-built; the other runs against systems that are already in production.

Regulation is increasing the demand for these records. The EU AI Act requires event logging, human oversight, and record-keeping along the value chain (Arts. 12, 14, 25), without naming a DBOM, and a DBOM is one credible way to package that evidence. Nothing that follows argues the DBOM is a bad idea. The finding is narrower: it is a genuine advance that is necessary and not self-certifying.

Applied to AI decisions, the lesson can now be stated precisely. A DBOM cannot tell you a recorded decision was justified, and nothing yet carries that judgment the way VEX carries exploitability. A component can be present and unexploitable. A decision can be right and under-supported. Correctness is measured against the outcome; supportedness is measured against the record, to the standard the function sets. The tempting dismissal, that a probably-right decision excuses a thin file, has it backwards. A correct outcome does not establish that the record was sufficient, just as a complete record does not establish that the decision was right. Supportedness is what lets anyone evaluate the call after the fact, and a record that cannot justify a correct call cannot flag an incorrect one either. The inventory shows the thing; it cannot grade it.

FIGURE 1

## Inventories describe. Verdicts decide.

*Knowing what exists is different from knowing whether it matters.*


**The inventory shows the thing. It cannot judge it.**

Four questions clarify what a decision record can establish. They are not four stages in a process, but four distinct properties a reviewer may test when deciding what the record can actually support.

**Integrity** asks whether the record was altered after it was made. Signing and attestation answer it. What they establish is the provenance and intactness of the record; the recorded conclusion remains as true or false as it was before anyone

signed it. Attestation proves what ran where, and stays silent on whether the output was right. A record built from an agent's self-report inherits a deeper problem: models have been shown to behave differently when they believe they are observed, so the agent's account of what it did is itself a claim needing corroboration.

**Completeness** asks whether the required fields or events are present. A schema answers it, and this is precisely what Microsoft's 0.0-to-1.0 score measures. The schema stops there, at presence; the question of what the assembled fields prove is left standing.

**Traceability** asks whether the path from inputs to output can be reconstructed. A well-kept thread of receipts answers it. Kroll's analysis of traceability in the FAccT literature draws the boundary exactly: a mechanical input-to-output trace does not explain a system's origins or why it behaves as it does. Reconstructing the path is a different achievement from explaining the decision, and different again from showing the decision was supported.

**Justification** asks whether the recorded facts are enough to support the outcome. That judgment weighs the basis behind the reasoning, the permission behind the access, and the authority behind the action. A recorded approval is a fact; the authority it carried is a further question, one the record may or may not be able to answer. A faithful explanation can help a reviewer weigh it; explanation and authorization stay separate all the same.

None of the four questions is answered by the record alone, because the record is not where the proof is made. Underneath any DBOM sits what the Evidence Debt framework calls the evidence plane: the systems that produce proof as a by-product of doing their work: enforcement points that emit logs, policy engines that record their grants, attestation services. The DBOM packages references to that proof. The two layers come apart in both directions. A plane can run without anyone assembling a DBOM from it, and a DBOM can be dutifully assembled from a plane too impoverished to prove anything.

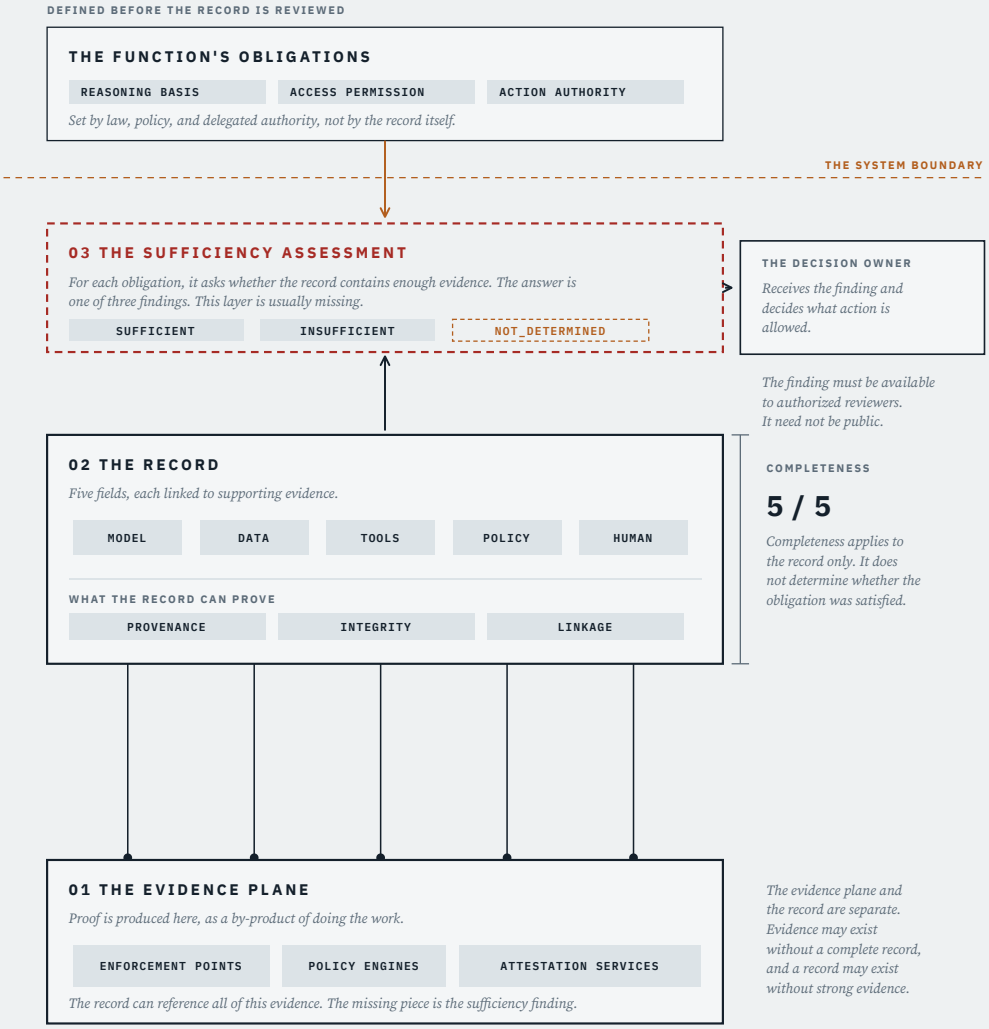
So a DBOM can satisfy its schema completely and still carry evidence debt, meaning obligations the record cannot discharge. The missing proof may be evidence the system could have captured and did not, a required condition the workflow never created, or a matter no record can conclusively settle. These three are different in kind, and a completeness metric cannot tell them apart. If a

complete DBOM reliably answered every material question of reasoning basis, access permission, action authority, and meaningful human control, no separate assessment would be needed. The next question is how to surface the shortfall without collapsing it into a score.

FIGURE 2

## Records are not introspective.

*Logs and audit trails describe events. They carry no inherent judgment about whether those events are sufficient for the mission use case.*




**Evidence debt begins when the record, complete or not, cannot satisfy the obligation.**

PART II

# From Completeness to Sufficiency

The evidence required of a decision is set by the function being performed: credit denial, medical triage, autonomous action in a shared space; the record format comes later. A consequential function usually carries at least three obligations. The reasoning must have had an adequate basis. The access must have been permitted. The action must have been within authority, including genuine human control where the function demands it.

Each obligation stands on its own. Strong evidence in one area does not make up for missing evidence in another. The prior essay stated the rule: evidence debt does not net, and a surplus on one question cannot pay a gap on another. The instrument that honors it follows the same division of labor established by SBOM and VEX:

> **The assessment reports each obligation on its own terms, SUFFICIENT, INSUFFICIENT, or NOT DETERMINED. What a finding should trigger belongs to whoever owns the function's risk.**

Two of these values sound similar, but the distinction between them matters. INSUFFICIENT means the evidence is in and fails the requirement. NOT DETERMINED is a finding about the file: the record cannot show whether the obligation was met, and both possibilities stay live. The world may have been fine; the record cannot say so.

A composite score will not work here. It can hide a failure in one obligation behind strong evidence for another. It also turns assessment into a decision once someone sets a threshold for what counts as acceptable. The approach below keeps those

functions separate: one tier checks the record, another assesses what it supports, and the final decision remains with the responsible person.

**Evidentiary obligations.** These are the questions used to assess the decision record. Each is evaluated separately. There is no composite score:

| OBLIGATION | ASSESSMENT |
|---|---|
| **Reasoning basis** | SUFFICIENT / INSUFFICIENT / NOT DETERMINED |
| **Access permission** | SUFFICIENT / INSUFFICIENT / NOT DETERMINED |
| **Action authority** | SUFFICIENT / INSUFFICIENT / NOT DETERMINED |

**Record assurance.** Questions about the record itself, narrower and more technically bounded, though not always easy in distributed pipelines:

| FIELD | RESULT |
|---|---|
| **Integrity of record** | verified / unverified |
| **Declared gaps** | explicit list |

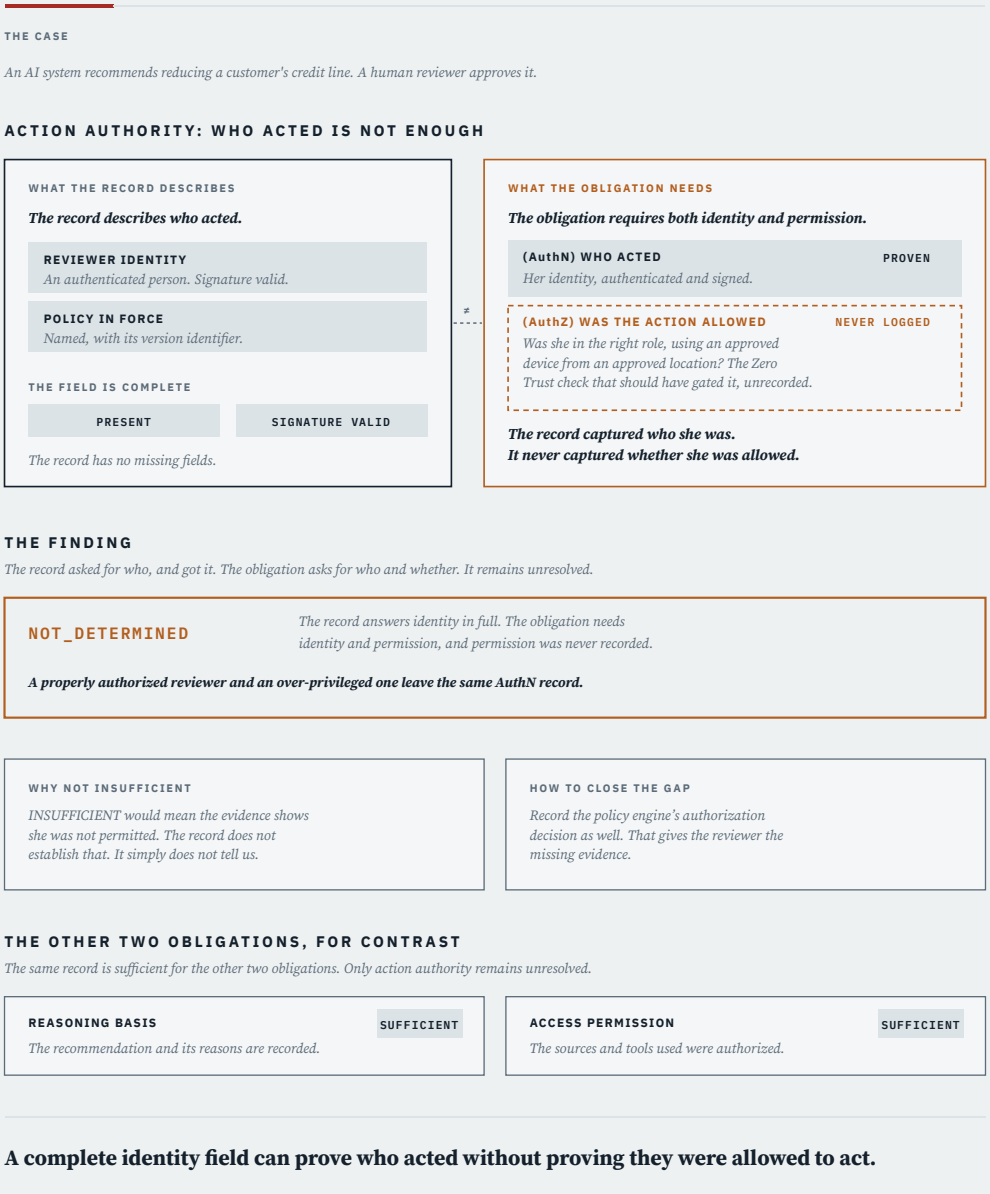
**The handoff.** The report stops there. The proceed, constrain, or prohibit call depends on the function's consequence and the mission's risk appetite, both outside the record, and it belongs to whoever owns that risk, the way the ship-or-not call on a VEX statement belongs to whoever reads it.

Microsoft's completeness score and the proposed assessment answer different questions. A Microsoft completeness score of 1.0 means all five schema fields were recovered from telemetry, a real accomplishment about the record. Whether the decision was good is a question the completeness score does not ask. Whether the record contains enough evidence to support that decision is a question the score cannot answer. A field can be present while its obligation reads NOT DETERMINED; Part III opens exactly that case. A completeness score can show that part of the record is missing, but it cannot explain why. The evidence may never have been captured, the system may not have been designed to preserve it, or the obligation may be impossible to prove after the fact. Those are different failures and require different responses.

FIGURE 3

## The Proof is in the Record?

*Records capture events. Evidence obligations determine whether those events prove what matters.*




Mechanical defects are easier to detect. A missing field, failed hash, or deleted object can be identified without understanding the underlying credit decision. But detecting the defect does not determine what the decision record still supports.

A DBOM should also declare its gaps. It should tell a later reviewer what the record does not establish, because silence can easily be mistaken for coverage. It should also distinguish between a check that was never performed and a check that was performed but found nothing. The academic DBOM threat model already makes a similar distinction by stating what the design protects against, what it can detect, and what it cannot address.

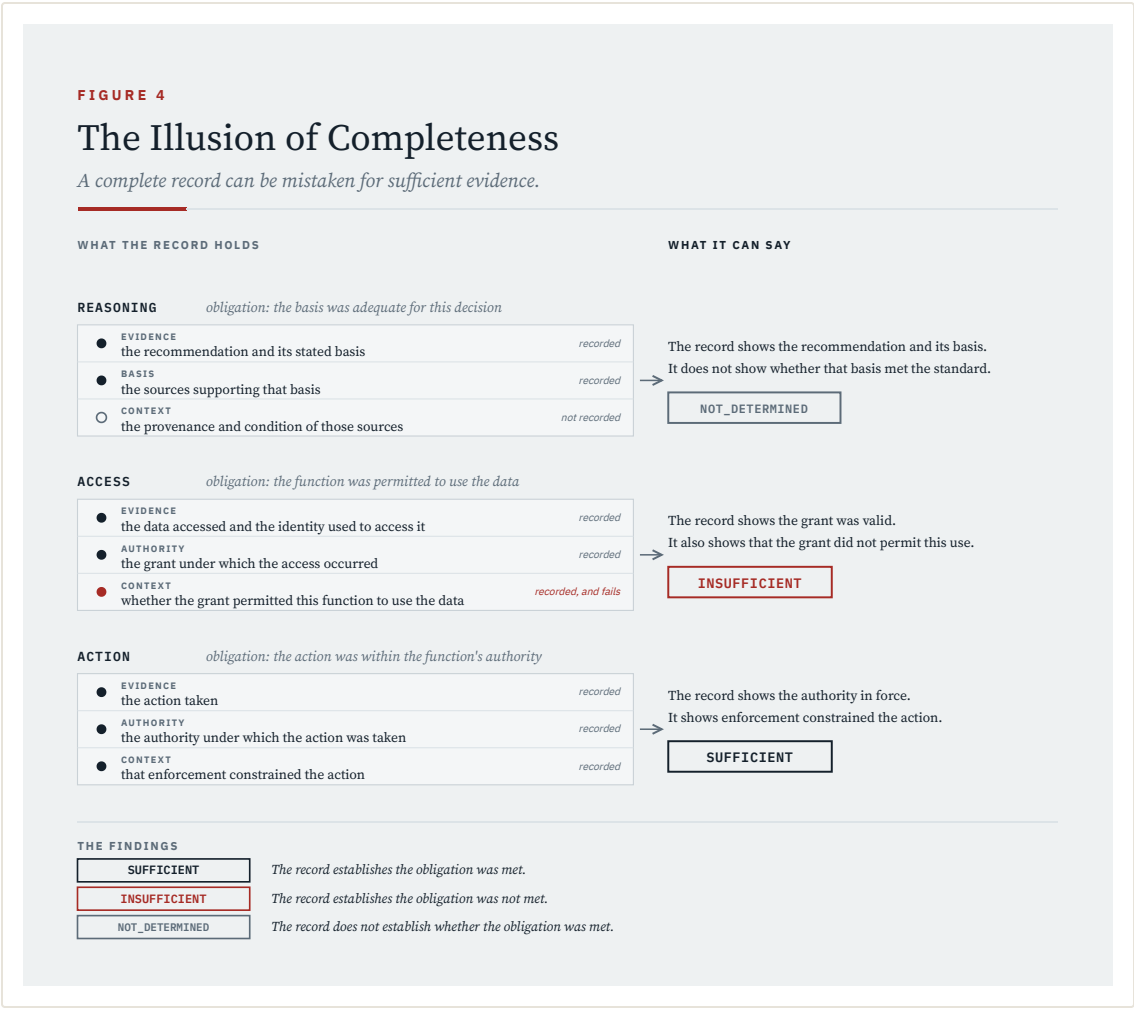
PART III

# The Value of an Approval

A signed approval records that a person approved the decision. To understand what that approval establishes, the record also has to show what the reviewer knew, what authority they held, and whether they could genuinely stop the action.

Consider a credit-line reduction recommended by an agentic system. Company policy requires a human reviewer to approve the change before it takes effect. If the customer later asks why her credit line was reduced, the institution must provide specific reasons for the adverse action. Under ECOA and Regulation B, saying only that a model score fell below a threshold is not enough (12 CFR 1002.9(b)(2)). The CFPB has also made clear that using a complex model does not excuse the institution from explaining the decision (Circular 2022-03).[1]

The DBOM records the recommendation, the reasons presented to the reviewer, the reviewer's identity, the approval time, the applicable policy version, the approval, and the resulting account change. All required fields are present, and the record receives a completeness score of 1.0. It answers every question about the approval except the one the customer is asking: whether the decision was adequately supported.




The remaining gaps fall into the three categories introduced in Part I:

| GAP TYPE | MEANING |
|---|---|
| **Capture gap** | The system could produce the evidence but currently does not. |
| **Control-design gap** | The workflow or authority model does not create the required condition. |
| **Inherently contestable** | No record can conclusively establish the underlying human or normative state, though indirect evidence can bear on it. |

Four questions show how differently the gaps behave. None of the four is the reviewer's fault: the case landed in her queue, she held the entitlement as far as anyone knew, and the system let the change through.

> **Was the reviewer authorized to approve this decision? The record identifies her, authenticates her, records her signature, and names the policy in force. But it does not contain the authorization decision made at the time of approval: which identity, device, location, network, and session conditions were evaluated, which policy rules applied, and whether the policy engine returned ALLOW.**

That distinction matters. The reviewer may have been the right person and the decision may have been part of her job, but the business may still prohibit that action from an unmanaged laptop, an untrusted network, or an offsite location. A signature proves who acted. It does not prove that the action was permitted under the conditions in which it occurred.

Because the record contains no contemporaneous authorization result, the assessment is NOT DETERMINED. This is a capture gap. Policy engines already make these decisions; preserving the inputs, policy version, and ALLOW or DENY result would close it. The completeness score does not reveal the gap because the authority field is present even though the authorization evidence is not.

**Did the reviewer see the relevant evidence?** The record shows the recommendation and its reasons appeared on a screen. Whether the reviewer opened the underlying account history, the uncertainty estimates, and the one signal that cut the other way is observable: a system can log which documents were opened. Whether she actually read or understood them is not. Most systems do not capture even the first. That is the second capture gap, closable at the ordinary price of instrumentation and retention.

**Could the reviewer have said no?** The answer lives in the design of the workflow, and no quality of the person in the chair changes it. The record can hold real evidence: the grant that defined the role, a refusal path, and above all whether refusing, when someone tried it, stopped the change. But suppose the reduction was queued before the reviewer ever saw it. Suppose that above a confidence threshold the workflow offered no way to reject, only to record that the

recommendation had been seen: a workflow with no terminal rejection state, whose telemetry shows nothing but valid events. No added schema field repairs that: the condition the field would attest to never existed, and the record can only report what the workflow made true. This is a *control-design gap*, and the fix lives in the authority model. From inside the chair, a decorative refusal path and a real one are indistinguishable, right up until the day someone refuses and finds out.

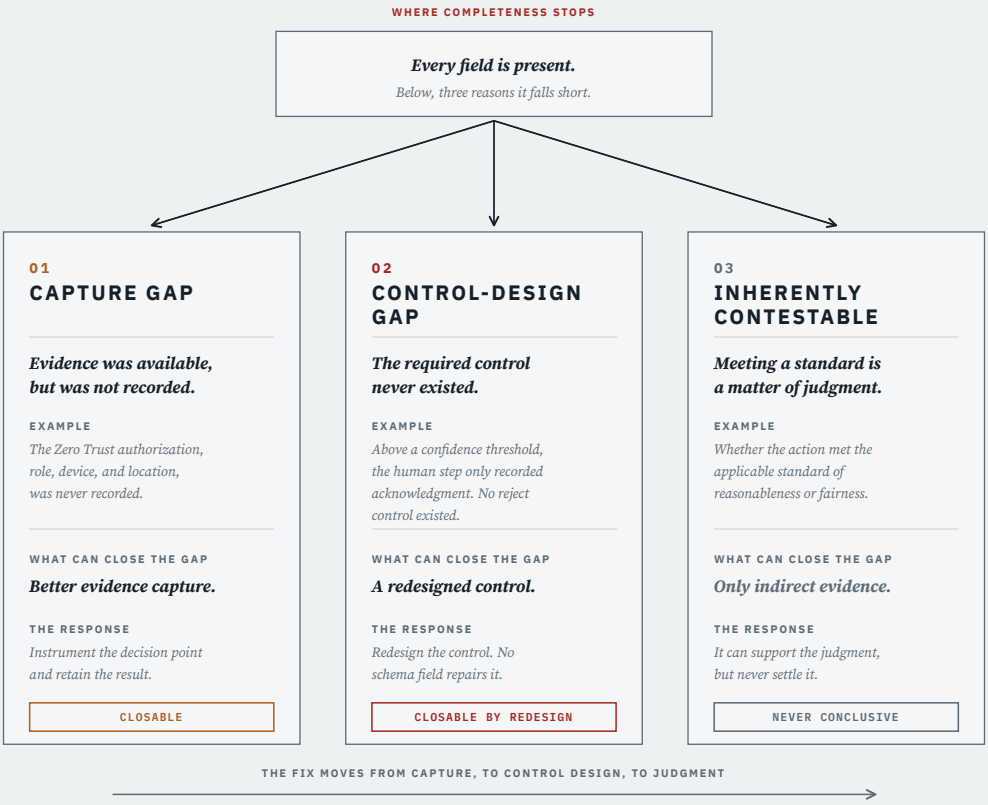
**Did the reviewer understand the uncertainty, or only ratify the recommendation?** No record settles what a person understood at 4:59 on a Friday. This one is *inherently contestable*. Indirect evidence still bears on it: what uncertainty was actually displayed, whether the workflow required a comprehension step, whether this reviewer has ever once diverged from the system's recommendation. It is not a story about careless people: automation bias persists even where oversight is legally mandated, the EU AI Act names it in statute (Art. 14), and repeated approvals degrade through the normalization of deviance. Capture alone cannot close this gap. It can only keep the contest honest.

Three checks remain, adequate time, action before the approval, and provable prevention of an unauthorized action; they sort into the same types, and Figure 5 carries them.

FIGURE 5

## Mind the Gap

*The remedy depends on why the record falls short of the obligation.*




The taxonomy prevents two opposite mistakes. An inherently contestable gap is not permission to record nothing, and a capture gap is not proof the evidence is worth its retention cost. The record may prove that the reviewer clicked approve, and still not give the customer a defensible explanation of why the reduction was justified, or show that anyone could have stopped it. **The record proves that approval occurred; it cannot show whether human authority stood behind it.**

**Each obligation splits into two questions. "What happened?" is empirical, and a sufficiently accurate record can answer it. "Was it legitimate?" requires comparing them against an external standard: law, policy, or delegated authority.**

**A DBOM can carry the facts and the relevant policy references; it cannot perform the comparison.**

**A DBOM is naturally strongest at recording facts about what happened. It can also package evidence that bears on legitimacy, such as policy identifiers, access grants, and authorization or attestation events. But it cannot package evidence that was never produced. When the surrounding systems never record the authorization decision, the reasoning basis, or other required evidence, the DBOM has nothing to preserve. Those unresolved obligations are evidence debt.**

CLOSING

# The Boundary of the Artifact

A DBOM can preserve, connect, and authenticate the evidence available to a later reviewer. It cannot establish, by schema conformance alone, whether that evidence supports the decision to the standard the function sets.

> **A DBOM is a container for evidence, not a certificate of legitimacy.**

The companion the field is missing does for the decision record what VEX does for the software inventory. It reaches a finding, obligation by obligation, on whether the recorded evidence supports the decision to the standard the function sets, states the record's declared gaps among those findings, and hands them to whoever owns the risk. Software supply-chain practice already makes this handoff through VEX, a structured mechanism the field uses for exactly this purpose. A decision, right or wrong, has earned the same companion. That is a direction, and another essay.

---

## NOTE

1 **The evidentiary duty may attach earlier than the approval step. In *SCHUFA* (C-634/21), the CJEU held that generating the score is itself decision-making under GDPR Art. 22 where a third party leans heavily on it, which places the duty at the moment the recommendation is produced.**