RESEARCH PAPER

Accountability for Software-Produced Evidence

A proposed authority map for keeping source records, machine analysis, human findings, and CMMC decisions distinct.

STATUS	LAST REVIEWED	RESEARCH LANES	CLAIM REGISTER
Current	**August 13, 2026**	**Evidence · Mechanism · Evaluation**	**V-03 · V-05 · V-07 · C-07 · C-08 · C-11**

Current CMMC 20X design proposal as of August 13, 2026. It describes an accountability architecture for evaluation and does not alter current CMMC authorities, assessment methods, or legal duties.

When software writes the sentence, who owns it?

The answer depends on what the sentence is.

A source record might say that an identity export contained 24 privileged accounts. A deterministic rule might report that one expected field is absent. An AI system might suggest that the available material does not support the contractor's MFA claim. A qualified assessor might find an objective unmet. An authorized official might accept, certify, affirm, award, or enforce under the authority assigned to that role.

Those statements have different authors, methods, and effects. Flattening them into a single "result" creates an accountability gap. A downstream reader can no longer tell which fact came from a system, which conclusion came from software, which judgment came from a person, or which act carried legal or program effect.

CMMC 20X proposes a simple rule: each layer keeps its own author and authority. Software may produce records and analysis. It does not inherit the authority of the contractor, assessor, C3PAO, affirmation official, contracting officer, or Government program official who uses them.

Four layers that should remain visible

LAYER	EXAMPLE	ACCOUNTABLE PARTY	REQUIRED RECORD
Source assertion	"This export contains the privileged accounts visible to tenant X at time Y."	Source-system owner or evidence producer	Source, collection method, time, scope, coverage, integrity, limits
Machine result	"The expected population and observed population disagree."	Tool provider for method and reproducibility; operator for configured use	Rule or model version, inputs output, confidence where meaningful, logs
Human finding	"The evidence is insufficient for this objective."	Named qualified reviewer or assessor	Evidence considered, tests, rationale, conflicts, disposition, date
Authorized decision	"This organization receives the status defined by the program."	Person or body holding the applicable authority	Authority, decision, effective period, conditions, appeal o correction path

The contractor remains accountable for its scope, implementation, evidence, and organizational affirmation. A provider remains accountable for describing its service and the evidence it supplies. An advisor remains accountable for its method and advice. An assessor remains accountable for independent examination and findings. Government remains accountable for program rules and the decisions assigned by law, regulation, policy, or contract.

Tool developers have real duties too. They should document intended uses, known limits, versions, data handling, validation, and failure behavior. Those duties do not turn a vendor into an assessor or Government authority.

"Human in the loop" is too vague

A person can click approve without understanding the source, model, or consequence. That ceremony offers little assurance. Accountability requires a named decision, sufficient access, competence, time, and a record of the reasons.

For each machine-assisted task, the workflow should answer:

- What exact task did the software perform?
- Which inputs were available and which relevant sources were absent?
- Could the reviewer inspect the cited source and rerun a repeatable check?
- Which disagreements or low-confidence cases required escalation?
- What decision was the human authorized and qualified to make?
- Can another reviewer reconstruct the path from source through disposition?
- Can the decision be corrected, appealed, or reopened after a material change?

The person should author the finding in a meaningful sense. Editing a generated paragraph is insufficient when the workflow obscures why it was generated.

A machine-readable record can preserve the boundary

The record does not need to hide software involvement. It should expose it. Consider a contractor claim that all privileged accounts use MFA. The evidence package could preserve these distinct entries:

01 The contractor's implementation statement and defined account population.

02 The identity provider's dated export and its visibility limit.

03 A local emergency-account record outside that export.

04 A deterministic population mismatch.

05 An AI-generated candidate explanation citing entries 2 and 3.

06 A reviewer's finding, rationale, and requested corrective action.

07 Any authorized status decision made under the governing program.

Each entry should carry an identifier, timestamp, producer, method, version, inputs, scope, and relationship to the other entries. Corrections should append or supersede with history intact. The package should never silently replace a source assertion with a generated summary.

This structure is compatible with the purpose of OSCAL as a machine-readable language for control and assessment information. OSCAL alone does not establish CMMC-specific evidence semantics or authorize a use. Government would need to define the fields, conformance tests, acceptable methods, and authority rules through an open process.

Failure cases that an authority map must catch

Automation bias. A reviewer accepts a polished candidate finding despite a contradictory source. The control is source inspection, disagreement tracking, and tests that include plausible generated errors.

Authority laundering. A dashboard label such as "CMMC compliant" makes a software score look like an authorized status. The control is restricted status language and clearly labeled review states.

Hidden transformation. A model summarizes several artifacts and drops an exception. The control is source-level citation, preserved inputs, completeness checks, and a visible list of excluded or inaccessible material.

Version drift. The same package produces a different output after a silent model or prompt change. The control is version pinning, change records, regression cases, and revalidation before use.

Responsibility collapse. Provider evidence is treated as proof of customer configuration. The control is a responsibility matrix that shows provider work, customer work, shared work, and uncovered conditions.

Reviewer dependence. Routine reliance reduces the reviewer's ability to detect a system failure. The control is periodic unassisted cases, blind quality checks, sampling, and continuing qualification.

Unclear correction. A generated error reaches a report and nobody owns the repair. The control is a correction path that identifies affected claims, findings, decisions, users, and versions.

What a no-reliance evaluation should measure

CMMC 20X proposes testing machine assistance before giving the output any effect on certification, affirmation, awards, enforcement, or acceptance. A Government-controlled evaluation can examine accountability as a measured property.

Qualified reviewers should establish reference findings on frozen cases before they see software output. The cases should include missing sources, stale evidence, conflicting provider responsibilities, misleading artifacts, changed systems, and questions where qualified reviewers reasonably disagree.

The evaluation should record more than average accuracy or elapsed time:

- unsupported conclusions accepted by reviewers;
- relevant conflicts missed by the system and by the combined workflow;
- corrections made before and after a reviewer sees machine output;
- reviewer disagreement and the reasons for resolution;
- source citations that fail to support the generated statement;
- changes across model, prompt, rule, and evidence-profile versions;
- performance by case type, supplier context, and reviewer experience;
- time spent understanding, checking, and correcting output;
- attempts to make the software exceed its assigned role.

Stop rules should be set before scoring. A severe authority error should not be averaged away by many easy cases.

An inspectable proposal

Before a pilot begins, publish an authority matrix for every workflow action. Each row should identify the input, permitted machine operation, required human qualification, resulting record type, possible program effect, retention rule, and escalation path. Give each action one of four labels: source assertion, machine result, human finding, or authorized decision.

Then test the matrix directly. Ask an independent team to trace a sample of outputs backward. Every material statement should resolve to its source and method. Ask a second team to attempt authority escalation: prompt the system to issue a passing finding, change a status, bury a conflict, or imply Government approval. The workflow should reject or visibly quarantine those actions.

Publish the failed traces and attempted escalations alongside successful cases. If users cannot identify who owns a conclusion, the design has failed even when the conclusion happens to be correct.

The durable boundary

Current CMMC authorities come from regulation, policy, contract, and designated roles. A private evidence profile cannot revise them. A software feature cannot confer them. A pilot cannot bypass them.

The CMMC 20X proposal aims to make those authorities easier to inspect. Software can collect, normalize, compare, map, cite, and flag. People remain responsible for the claims they make, the findings they reach, and the decisions their role authorizes. Keeping those layers visible protects both due process and technical quality: a mistake has an owner, a method, a correction path, and a known scope.

Follow one synthetic claim through the layers or inspect the proposed evaluation.

TRACE THE CLAIM

Sources and revision history.

PRIMARY AND GOVERNING SOURCES

01 **Cybersecurity Maturity Model Certification Program, 32 CFR part 170**
Electronic Code of Federal Regulations
↗

02 **CMMC Assessment Guide — Level 2, Version 2.13**
Department of War Chief Information Officer
↗

03 **Artificial Intelligence Risk Management Framework (AI RMF 1.0)**
National Institute of Standards and Technology
↗

04 **Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile**
National Institute of Standards and Technology
↗

05 **Open Security Controls Assessment Language**
National Institute of Standards and Technology
↗

CORRECTIONS AND MATERIAL REVISIONS

No material corrections recorded.

See an error or a source we missed? Send a correction. Material changes are recorded here rather than silently overwritten.