Preprints.org

Article

Not peer-reviewed version

Propagating Corrections in AI Assisted Organizational Knowledge Systems

Connor Nitchals *

Posted Date: 21 October 2026

Keywords: knowledge management; retrieval augmented generation; data provenance; knowledge revision; generative artificial intelligence; organizational learning; human authority

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC, OpenAlex.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Propagating Corrections in AI Assisted Organizational Knowledge Systems

Connor Nitchals

Independent Scholar, United States; c_nitchals@mail.fhsu.edu

Abstract

Organizational knowledge systems increasingly use generative artificial intelligence to transform source documents into answers, summaries, and reusable guidance. When a source changes, revising that source alone can leave dependent content available for further use. This article develops a theoretical framework for governing such corrections. It combines a focused analysis of knowledge management, data provenance, truth maintenance, and retrieval augmented generation with a formal model of versioned claims and declared support. The model separates source acceptance, semantic certification, service eligibility, and the reopening of past decisions. Two propositions establish the scope of dependency traversal and the monotonic effect of withdrawing accepted support under fixed assumptions. A further impossibility result identifies a limit on targeted notification when material dependencies are missing. A constructed reimbursement example demonstrates how alternative support can preserve a valid exception while unsupported guidance is quarantined. The article proposes an epoch based correction protocol, measures of notification coverage and stale exposure, and an evaluation design for future implementation. The contribution is a specification of correction obligations across knowledge lifecycles, rather than a new retrieval algorithm or an empirical performance claim. Effective correction requires both traceable dependencies and an institutional capacity to suspend reuse and review consequences.

Keywords: knowledge management; retrieval augmented generation; data provenance; knowledge revision; generative artificial intelligence; organizational learning; human authority

1. Introduction

An organization may correct its policy manual while continuing to distribute an assistant answer based on the previous version. The answer may have been copied into a frequently asked questions page, summarized in a training note, or used to justify an administrative decision. Each object has a different owner and update cycle. A correction to the manual therefore creates a question that document editing alone cannot answer: which other objects must be examined, and what evidence establishes that their continued use is justified?

This question concerns the maintenance of organizational knowledge. Established knowledge management research connects information technology to knowledge creation, storage, transfer, and application [1]. Organizational knowledge also depends on interpretation and social practice [2]. A technically correct update can consequently remain incomplete when the people or systems relying on an earlier representation continue to act on it.

The Problem of Dependent Reuse

Retrieval augmented generation combines a language model with access to an external collection of information [3]. This architecture creates an identifiable place to update source material. It also creates additional objects whose dependencies can outlast a retrieval session. Generated answers may become records, examples for staff, or inputs to later generation. A cache may preserve an old answer after the retrieval index has been rebuilt. A human decision may remain effective after the evidence

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

cited in its explanation has been withdrawn. The relevant unit of maintenance is therefore a dependency extending across objects and actions.

An attached citation is useful but insufficient for this task. Research on citation generation distinguishes answer correctness from the quality and completeness of citation support [4]. A citation can identify a relevant document without identifying the version, passage, scope, or combination of premises that justified a particular statement. It also need not identify later uses of that statement. The present article addresses these missing connections without treating citation, provenance, and justification as interchangeable.

The central question is: under what conditions can an organization justify the claim that a correction has reached the dependent uses within a declared system boundary? Three subsidiary questions organize the analysis. How should support be represented so that repeated copies do not appear to provide independent evidence? When should a correction suspend service, preserve an independently supported claim, or reopen a past decision? What can an institution establish when its dependency records are incomplete?

Contribution and Boundaries

The article develops a versioned claim model, three conditional analytical results, and a correction protocol that connects technical dependencies to institutional responsibilities. It also provides a worked example and an implementation evaluation design. Its proposed contribution is the combination of scoped support, explicit service restrictions, correction coverage, and consequence review within one specification for organizational knowledge maintenance.

The component ideas have substantial precedents. Database provenance identifies source contributions [5,6], and truth maintenance systems track the consequences of revising assumptions [7,8]. The article does not claim to originate dependency traversal, Boolean justification, or belief revision. Its narrower intervention concerns how these resources can specify correction duties when generated material becomes persistent organizational guidance.

The analysis concerns finite, instrumented workflows in which claims can be associated with identifiable versions and declared dependencies. It does not provide a method for discovering every dependency inside a language model. It does not infer the truth of a claim from the existence of a provenance record. Its numerical illustration uses stipulated quantities, and its recommendations remain subject to empirical and organizational evaluation. These boundaries are essential because a complete technical trace and an adequate institutional response are distinct achievements.

2. Conceptual Foundations and Related Work

Knowledge Maintenance and Provenance

The knowledge management perspective makes correction a lifecycle activity. A source can be acquired accurately and become unsuitable through a change in policy, evidence, jurisdiction, or intended audience. Retrieval can remain successful in the narrow sense of returning a relevant passage while application becomes inappropriate. Maintaining knowledge therefore requires retaining the conditions under which an object was accepted, as well as its content. The relevant organizational question is whether the representation remains warranted for the use now being made of it.

Provenance supplies a vocabulary for these conditions. Buneman and colleagues distinguish the contribution of source data to an output from the locations from which output values were taken [5]. Provenance semirings provide a richer algebraic treatment of alternative and combined contributions [6]. These distinctions matter because an answer supported by either of two sources should respond differently to a withdrawal from an answer requiring both sources. A flat list of references obscures that difference.

The W3C PROV data model represents entities, activities, agents, derivation, and responsibility, and includes mechanisms for recording revision and invalidation [9]. These concepts support

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

interoperability between systems that otherwise maintain incompatible histories. However, representing that one object was derived from another does not, by itself, specify when the dependent object must be blocked, who can authorize its return to service, or whether a past decision should be reconsidered. Those are additional policies that can be expressed alongside provenance.

Cheney and colleagues distinguish several forms of database provenance and their uses [10]. Herschel and colleagues further organize provenance research around purposes, representations, and deployment constraints [11]. Following that literature, the present model makes its purpose explicit: identifying potentially affected uses and checking whether they retain accepted support after a correction. It does not attempt to replace a general provenance standard or choose one representation for every application.

The distinction between process history and justification remains decisive. A language model can use a document and misrepresent it. An editor can correctly paraphrase a passage while overlooking an exception elsewhere. A trace can faithfully record either process. The proposed framework therefore attaches semantic certification to support records. Certification records a review of the relationship between a claim, its premises, and its intended context. It is a fallible institutional judgment with an owner and a version, rather than a guarantee supplied by the graph.

The framework also preserves unsuccessful and disputed histories. Deleting an outdated answer would prevent future retrieval but could remove the evidence needed to explain a previous decision. Separating the retained record from its permission for current service allows an organization to preserve accountability while stopping reuse. Archival availability and operational eligibility consequently require different status fields and access rules.

Truth Maintenance and Generative Systems

Truth maintenance systems supply a direct precedent for recording justifications and revising dependent beliefs [7]. Assumption based truth maintenance represents alternative environments in which a conclusion can hold [8]. The present support model uses a deliberately limited, positive fragment of that tradition. Accepted support may be combined conjunctively within a bundle or disjunctively across bundles. It excludes default negation, priorities between conflicting rules, and automatic conclusions from the absence of a record. Those exclusions make the conditional monotonicity result transparent.

Generative systems introduce a different difficulty: the dependency relationship may be uncertain or incompletely captured. A survey of hallucination research documents the broader problem of generated content that departs from source material or factual grounding [12]. Self RAG adds retrieval and self reflection to generation [13], while Corrective Retrieval Augmented Generation evaluates retrieved material and can trigger further retrieval [14]. These approaches address the production of better grounded responses. They do not make a record of an earlier response disappear when an organization revises a source.

Work on source reliability is also relevant. Hwang and colleagues estimate source reliability and use it in retrieval and aggregation [15]. Reliability estimation addresses differences among evidence providers. A correction protocol additionally needs to identify objects already distributed and the conditions attached to their earlier approval. High source reliability cannot establish that a particular version is current for a particular context.

Knowledge editing provides a closer comparison. MQuAKE evaluates whether changes to facts produce appropriate changes in answers requiring related facts [16]. Work on ROME studies targeted changes to factual associations within model parameters [17]. These contributions demonstrate why a local edit and its wider implications must be evaluated separately. The present proposal addresses explicit organizational artifacts and decision records; it offers no guarantee that editing an external source changes a model's internal associations.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

Table 1. Analytical approaches and the correction task addressed here.

Approach	Primary unit	Useful contribution	Remaining task
RAG and citation evaluation	Retrieved evidence and answer	Assess grounding at generation	Track persistent reuse
Data provenance	Source contributions and activities	Record origin and derivation	Specify service restrictions
Truth maintenance	Beliefs and assumptions	Revise dependent justifications	Allocate institutional duties
Knowledge editing	Model facts and related answers	Evaluate implications of edits	Review distributed artifacts
Proposed framework	Versioned claims and uses	Connect support to correction duties	Validate capture and response

The comparison in Table 1 identifies complementary levels of analysis. A system can combine retrieval evaluation, source reliability estimates, explicit dependency records, and human review. The claim here is that the correction duties across these levels should be specified and tested separately. Describing a system as corrected without declaring which level was addressed prevents a meaningful assessment of the claim.

3. Materials and Analytical Methods

Study Design and Source Selection

This is a theoretical design study. Its materials are published accounts of knowledge management, provenance, truth maintenance, generative retrieval, and AI governance. The literature is used to identify concepts, specify assumptions, and compare the proposed framework with established approaches. The study is neither a systematic review nor a meta analysis. It reports no search completeness estimate, pooled effect, participant sample, or deployed intervention.

Source selection was purposive and organized around five analytical functions. Knowledge management sources establish the lifecycle problem; provenance sources distinguish source contributions and process records; truth maintenance sources establish dependency based revision; generative AI sources identify grounding and editing problems; governance sources inform ownership and review. Publisher pages, proceedings records, author repositories, and standards documents were used to check bibliographic details and the claims attributed to these sources. Sources were selected for relevance to those functions, rather than to estimate how frequently a position appears in the literature.

The procedure contains four stages. First, the problem is decomposed into source change, dependency discovery, support reassessment, future service, and past consequence review. Second, each stage is assigned a distinct object or state in the formal model. Third, propositions are proved under explicit assumptions and examined through counterexamples. Fourth, a small constructed case instantiates the model and compares stipulated correction policies. The evaluation design in Section 7 describes future tests and is not presented as completed research.

Analytical Criteria

The model is assessed for internal consistency, traceability of its assumptions, and diagnostic usefulness. Internal consistency requires a clear distinction between accepted support and factual truth. Traceability requires each eligibility decision to refer to identifiable versions and an applicable context. Diagnostic usefulness requires the framework to distinguish a source replacement from a

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

completed downstream response. An adequate account must also explain why independent support can survive a correction and why missing records limit completeness claims.

The formal results establish properties of a representation and protocol, conditional on their assumptions. They cannot establish that actual staff have sufficient time, that a support extraction model captures every premise, or that an appeal procedure functions fairly. Such questions require observation and evaluation. Human AI interaction guidance provides a precedent for designing interfaces that make system behavior and correction intelligible to users [18], but a diagram of such an interface would not establish its usability.

All objects, durations, and exposure weights in the worked example are constructed for this article. They illustrate how the definitions behave; they are not estimates of a real organization's error rate or correction capacity. The example can be recomputed from its stated graph, support families, and tables. This choice permits an auditable conceptual comparison without implying access to confidential organizational records.

4. Formal Model of Supported Knowledge Objects

Versions Contexts and Dependency Records

Consider a finite directed acyclic graph of immutable object versions at a declared snapshot. Vertices include source statements, extracted claims, generated answers, cached representations, and decision records. A directed edge from an upstream vertex to a downstream vertex records a declared material dependency. An edge means that the upstream object contributed to a stated justification or interpretation. A citation or a semantic similarity match alone does not establish an edge of this kind.

$$G_e = (V_e, E_e), \qquad S_e \subseteq V_e, \qquad E_e \subseteq V_e \times V_e. \tag{1}$$

A correction creates a new version and a status event for the earlier version. It does not overwrite the earlier object's identity. The graph remains acyclic because a newly generated version may depend on earlier versions, while a revised earlier object receives a new identity. Mutually reinforcing documents that cite one another can still exist in practice; their versioned histories are represented as successive events. Where a snapshot contains unresolved circular justification without an external warrant, the system cannot certify that cycle merely because every vertex has an incoming edge.

A context specifies the population, task, location or institutional domain, and applicable time interval for which a claim is being considered. It can include a policy category or a document access class. Two identical sentences can have different eligibility in different contexts. The statement that claims must be filed within thirty days may remain suitable for a previous cohort while being superseded for new entrants. Context matching is therefore a condition of support, not a descriptive tag added after a response has been approved.

Table 2. Minimum record fields for an inspectable knowledge object.

Field	Record	Purpose
Version identity	Stable identifier and content digest	Distinguish revised objects
Context	Population, task, domain, time	Prevent overbroad application
Dependencies	Material inputs and capture method	Discover candidate uses
Support and certificate	Alternative bundles and scoped review	Check eligibility
Ownership and status	Custodian, downstream owner, service state	Assign and enforce response

The graph is a declared representation. An institution should distinguish edges captured directly during a workflow from edges later inferred by a model or asserted by a reviewer. Inferred

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

edges may be useful for discovering candidates, but they require confidence and review metadata. Treating all inferred links as established justification would shift uncertainty into the trace while presenting its output as exact.

At each snapshot, a live object is one available for current operational use. A retained decision record can be in the graph without being a live answer. This distinction prevents an instruction to suspend current guidance from being interpreted as an instruction to delete historical evidence. The subsequent definitions evaluate support for a context and a snapshot; they do not assign a permanent truth value to the text.

Accepted Support and Service Eligibility

Let the primitive warrants be source versions or explicitly reviewed assumptions that terminate the declared justification. A primitive warrant may be an approved policy statement, a validated observation, or a versioned transformation rule. Primitive status marks the boundary of the representation, rather than absolute epistemic certainty. Every primitive has a custodian and acceptance conditions. An unrecorded assumption cannot be supplied by declaring that the output is generally plausible.

For each object, its support family contains the bundles of primitive warrants judged sufficient for the claim in its context. A bundle is conjunctive: every member must be accepted. Bundles are alternatives: one fully accepted bundle suffices for the support test. An object with no established support has an empty family. A source warrant has the singleton family containing itself. Empty bundles are excluded so that the model cannot manufacture unconditional support.

$$Q_e(v,c) = \bigvee_{B \in S(v,c)} \bigwedge_{s \in B} a_e(s,c). \tag{2}$$

The acceptance indicator is one only when the primitive's version, status, and scope satisfy the institution's declared criteria at the snapshot. The support test is separate from a certification indicator for each bundle. Certification records that a reviewer or validated procedure has checked the claim's interpretation, context, and particular support bundle, and that the certificate remains applicable. If a certificate binds to only one bundle, an alternative bundle cannot be used without a certificate covering it. Certification of several alternatives is possible but must be explicit.

$$Eligible_e(v,c) = \eta_e(v,c) \wedge \bigvee_{B \in S(v,c)} [\kappa_e(v,B,c) \wedge \bigwedge_{s \in B} a_e(s,c)]. \tag{3}$$

Here the contextual status indicator records whether the object is permitted for current service; the certification indicator is evaluated inside the disjunction for each bundle. Eligibility is a policy judgment about acceptable use under recorded conditions. A false statement can pass if its source and certificate are wrong. Conversely, a true statement can fail because it lacks an adequate warrant for an institutional decision. The model prevents these distinct questions from collapsing into a single confidence score.

Suppose a claim requires both a policy statement and a category rule. Its support family has one two member bundle. Withdrawing either primitive defeats that bundle. Suppose instead that two independently approved policies each suffice for the same scoped claim. Its family contains two singleton bundles. Withdrawing one can leave support intact. These examples explain why the graph determines which objects deserve examination, while the support family determines whether accepted support remains.

Copying a generated statement into a second document does not create an independent primitive warrant. If both copies ultimately depend on the same source, expanding their support expressions yields the same source identifier. The framework therefore blocks a simple form of apparent corroboration through repeated organizational reuse. It does not infer statistical independence between distinct sources; shared authorship, data, or policy origins may require further provenance analysis before alternatives are certified.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

Correction Reachability and Support Withdrawal

A correction event names a target version, the contexts affected, the reason for change, an event identifier, and an authority responsible for its disposition. The candidate set is the downstream closure of the target in the declared graph, excluding the target itself. Closure is conservative: it identifies objects whose justification may need reassessment. It does not imply that every candidate is false or that all candidates require the same response.

$$v \in D_e(r) \Leftrightarrow v \in V_e \wedge v \neq r \wedge r \rightsquigarrow v. \tag{4}$$

Proposition 1. *If every material dependency within the declared boundary is present in the finite graph, forward traversal from a corrected target discovers every object materially dependent on that target. The traversal examines the reachable vertices and their outgoing edges once when a visited set is used.*

Proof. Every dependent object has a finite directed path from the target by the completeness assumption. Induction on path length shows that traversal discovers the first successor and then each successive vertex. Conversely, each discovered vertex was added through an outgoing dependency edge from a discovered predecessor, so it lies on a path from the target. Marking a vertex visited prevents repeated expansion. The familiar linear bound in the visited subgraph follows. The result is a property of recorded reachability, not a guarantee that the graph captures semantic dependence in practice.

Proposition 2. *Holding context, support families, certification, and permission status fixed, changing any subset of primitive acceptance indicators from one to zero cannot change an ineligible object into an eligible object under Equations 2 and 3.*

Proof. A conjunction of positive acceptance indicators cannot become true when one or more inputs decrease. A disjunction of such conjunctions also cannot become true. The support indicator is therefore monotone in its primitive inputs. Holding each bundle's certification and the object's permission indicator fixed preserves this ordering in the eligibility expression. An eligible object can remain eligible through an unaffected alternative bundle or become ineligible; an ineligible object cannot acquire eligibility from withdrawal alone.

The qualification is substantive. A correction can introduce a new source, revise the claim, or change its context. Such actions alter the support family or other indicators and are outside Proposition 2. A default rule based on the absence of evidence would also fall outside the positive support language. The proposition should not be generalized to nonmonotonic legal reasoning, defeasible inference, or unrestricted natural language arguments.

A candidate can retain support while still requiring notification. Its wording may mention the withdrawn source, its explanation may need revision, or its users may reasonably need to know why a warrant changed. Reassessment must therefore distinguish loss of eligibility from the broader obligation to communicate a correction. The formal support test determines one part of the response; it does not exhaust institutional judgment.

Limits of Observation and Measures of Exposure

Proposition 3. *With incomplete dependency records and no additional information, a procedure cannot guarantee both complete targeted notification of all affected objects and zero notification of unaffected objects across every underlying graph consistent with the observed records.*

Proof. Consider observed records containing a corrected source and an object with no recorded path from that source. One possible underlying graph contains no such dependency. A second graph has an omitted material edge connecting the source to the object. The procedure receives identical

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

observed records in both cases. If it notifies the object, it produces an unnecessary notification in the first case. If it does not, it misses an affected object in the second. Randomization cannot guarantee both properties with certainty. Additional evidence or an explicitly broader alert policy is required.

The result leaves practical choices open. An organization may issue broad alerts, inspect likely unlogged copies, or restrict the reuse of material whose lineage is uncertain. Those actions can be reasonable without producing a claim of complete targeted correction. A report should state its boundary, missing capture channels, and the confidence attached to inferred dependencies. A small candidate set is not evidence of low impact when capture is incomplete.

For a specified event, define the unsupported live set as previously eligible live objects in the candidate set that fail the post event support test in their applicable contexts. Fix this set and its weights before comparing policies. Notification coverage records whether a correction notice reached the responsible owner or designated recipient. It does not record whether the object was revised or whether a past decision was repaired.

$$C_w(t) = \frac{\sum_{v \in U} w_v N_v(t)}{\sum_{v \in U} w_v}, \qquad \sum_{v \in U} w_v > 0. \tag{5}$$

In Equation 5, N is the binary indicator of delivered notification, indexed by object and time. Nonnegative weights represent use volume or a declared priority and require a rationale. If their sum is zero, coverage is not applicable. A separate completion measure can use verified disposition instead of notification. This distinction prevents a notification dashboard from being mistaken for evidence that unsupported service has stopped.

For future stale use, define exposure over a fixed horizon as the use rate multiplied by a consequence weight and the duration during which unsupported content remains available. The constant rate form is appropriate only when the example or measurement design supports that simplification.

$$H(T) = \sum_{v \in U} \int_0^T \lambda_v(t) c_v(t) I_v(t)\, dt. \tag{6}$$

In Equation 6, lambda is the use rate, c the nonnegative consequence priority per use, and I the indicator that an unsupported object remains available for service. Time starts at event registration. These quantities may vary within the horizon T. Priorities are normative choices. The measure concerns opportunities for new unsupported use; it excludes realized harm, interrupted access to valid guidance, and review costs. Lower stale exposure alone therefore does not establish the best overall policy.

5. A Protocol for Institutional Correction

Event Registration and Reassessment

The proposed protocol begins when an authorized correction event is registered. Registration records the contested version, applicable contexts, initiating evidence, responsible custodian, and the time at which restrictions become effective. Reports from ordinary users can initiate review without granting those users authority to withdraw arbitrary material. An intake process should preserve a challenge while separating allegation, provisional precaution, and confirmed revision.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

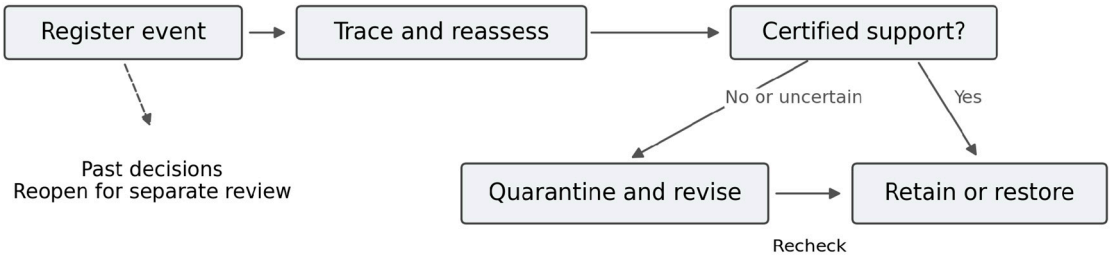

Figure 1. Proposed correction workflow. The past decision review branch remains open independently of current serving eligibility. Retention or restoration requires an applicable certificate.

The protocol assigns a new correction epoch to the relevant serving scope. A request can return an object only if its eligibility has been checked against the current epoch. This requirement applies to cached answers and exported reusable guidance within the declared boundary. An epoch label with no serving enforcement provides an audit marker but does not prevent stale use. Organizations must therefore identify the actual places at which content is delivered, including secondary applications.

Reassessment traverses the candidate set and evaluates the affected support expressions. A claim with no accepted certified bundle is quarantined for current service. A claim with an unaffected certified alternative can remain available, provided its displayed explanation is updated when necessary. A claim with uncertain semantic scope enters review. Revising an object creates a new version, records its warrants, and requires an applicable certificate before service resumes.

Table 3. Distinct dispositions after a correction event.

Condition	Current service	Further obligation
No accepted certified bundle	Quarantine	Revise or retire
Certified alternative survives	Retain within its scope	Update explanation if needed
Scope or support uncertain	Suspend affected use	Obtain semantic review
Past decision used old claim	Retain historical record	Review consequences separately

These dispositions separate two concurrent tasks. Technical controls stop further unsupported use. Organizational owners determine what should replace it and what past actions need attention. An answer can be quarantined quickly while its replacement takes longer to prepare. Recording both times permits an institution to report precaution and completed repair without treating them as the same event.

Serving Consistency Ownership and Closure

An implementation must handle requests that overlap a correction. One option uses a read barrier: a response prepared under an earlier epoch is rechecked before it is returned, and a cache entry is reusable only when its recorded epoch satisfies the current requirement. Another option blocks the affected scope until an atomic update of its serving metadata is complete. No response may use superseded acceptance metadata after the restriction is committed. The delay between event registration and effective enforcement remains a measured source of exposure; the consistency requirement does not make that delay zero.

A graph query and a cache update are rarely one indivisible operation. Failed workers, delayed index rebuilds, and disconnected services can leave inconsistent copies. The protocol therefore records acknowledgments from each controlled serving endpoint. An endpoint that cannot establish its current status is treated as unavailable for the affected scope. This is an availability tradeoff, not a

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

claim that every organization should stop all services whenever any source is challenged. Scope and proportionality must be designed before incidents occur.

Correction ownership has at least two functions. A source custodian can approve or withdraw a warrant. A downstream owner can review the use made of it and act on consequences. Internal algorithmic auditing research treats accountability as a lifecycle practice [19]. Evidence on data cascades also shows how upstream data problems can produce delayed downstream effects [20]. The proposed allocation of duties is an institutional design inference from those concerns, rather than an empirical finding that a particular allocation is effective.

Nitchals defines meaningful human authority through conditions including access to evidence, time, intervention power, contestability, and accountability traceability [21]. The present model applies that concern to correction work: a named reviewer must be able to inspect the relevant warrants, suspend unsupported reuse, and reopen a consequential decision. A dashboard that assigns an owner without providing these capacities does not establish a functioning review process.

Privacy and Preservation

Dependency records can reveal sensitive facts about who consulted a document, what decision was being considered, or which team questioned a policy. The graph should therefore store the minimum identifiers needed for correction, enforce access by role, and separate operational linkage from public explanations. A recipient may need to know that a justification was superseded without receiving another person's confidential case details. Provenance itself requires a retention policy and an accountable custodian.

Closure is recorded per obligation. A system may be closed for future serving while past decisions remain under review. Each endpoint needs a verified disposition, such as replaced, retained through independent support, archived, or transferred for consequence review. Notification acknowledgment alone is inadequate evidence of closure. A final event report states unresolved obligations and the system boundary within which completion was assessed.

6. Constructed Example of a Reimbursement Correction

Objects and Support Assumptions

Consider a fictional organization that reimburses staff training costs. An earlier policy version states that claims must be submitted within thirty days. A new approved version permits sixty days for a current cohort. A separate grandfather provision retains thirty days for a previous cohort. This example concerns an invented internal rule; it is not a statement about legal reimbursement requirements or any actual institution.

The earlier policy warrant is p. An independently approved grandfather warrant is q. The extraction x depends on p. The frequently asked questions page f and onboarding memo m depend on x. A saved assistant answer a depends on f, and a cached briefing b depends on m. An exception guide z has two separately certified alternatives: a derivation from p and a derivation from q. Decision records d1 and d2 record prior actions using a and b. The newly approved sixty day source is a replacement warrant with a new version identity and is not silently substituted for p.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

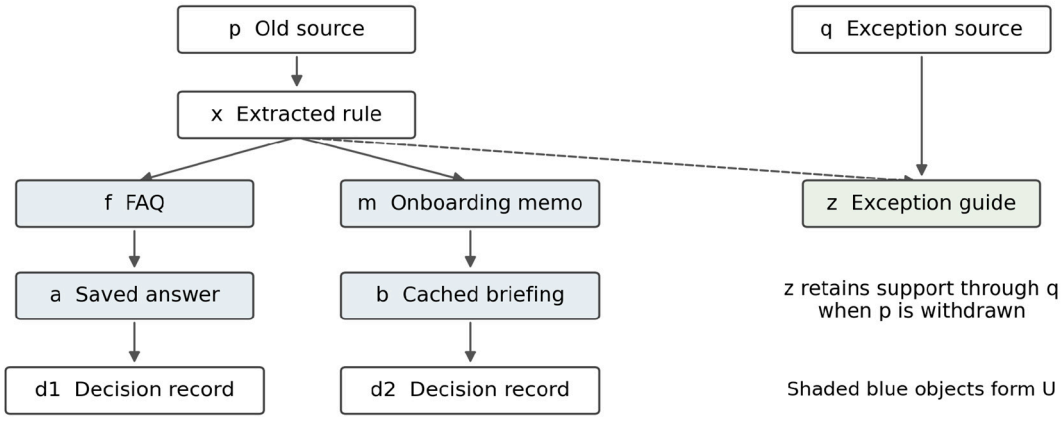

Figure 2. Declared dependencies in the constructed example. Arrows run from a material input to its dependent object. The dashed route to z is one alternative; q supplies the other. Decision records are reviewed separately.

Before the event, all five live guidance objects pass the support test in their respective contexts. At the event, p is withdrawn from current operational acceptance. Warrant q remains accepted for the previous cohort, and its certificate independently covers z. The support family of each of f, a, m, and b contains the single bundle consisting of p. The family of z contains two singleton alternatives, p and q. These stipulated families allow every support result to be checked directly.

The downstream candidate set includes x, f, a, m, b, z, d1, and d2. Its unsupported live subset contains f, a, m, and b. Object z remains eligible through q after its explanation is checked. The decision records are retained and sent to their owners for review; they are excluded from the future guidance exposure calculation. This exclusion avoids double counting the same assistant use as both an answer and a subsequent recorded action.

Stipulated Policy Comparison

Assign daily exposure weights of two to f, three to a, one to m, and four to b. A weight is the product of an assumed daily use rate and a consequence priority in arbitrary units. These values are chosen to show that a less visible cache can matter more than a frequently edited document. They are not measured use frequencies or estimates of actual harm. The unsupported set has total weight ten per day. Object z has weight two but remains supported and is excluded from this denominator.

Compare three stipulated policies over four days. The source only policy replaces p but leaves all four unsupported guidance objects available for the full horizon. The visible document policy notifies and replaces f and m after one day while a and b remain stale for four days. The full recorded dependency policy stops all four unsupported guidance objects after one quarter of a day. Its replacements can take longer; the assumed quarter day is time to effective quarantine, not time to complete substantive repair.

Table 4. Constructed comparison over a four day horizon. Times and weights are assumptions, not observed results.

Policy	Stale days f a m b	Notified weight out of 10	Coverage	Exposure units
Source only	4 4 4 4	0	0%	40
Visible documents	1 4 1 4	3	30%	31
Full recorded dependencies	0.25 for all	10	100%	2.5

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

The source only exposure is ten multiplied by four, or forty units. Under the visible document policy, f and m contribute three units during their first day, while a and b contribute twenty eight units across four days, totaling thirty one. Under full recorded dependency handling, ten multiplied by one quarter gives two and one half units. These differences follow from the stipulated service durations. They are not evidence that a proposed implementation will achieve a particular response time.

Weighted notification coverage is zero, thirty percent, and one hundred percent respectively for the unsupported live set. The corresponding unweighted figures are zero, fifty percent, and one hundred percent. The contrast shows why a coverage report should disclose both the denominator and any weighting. It also shows why correcting half the visible artifacts can leave most modeled exposure unresolved.

Sensitivity and Counterexamples

If quarantine takes two days, the full recorded dependency policy yields twenty exposure units. If it takes four days, its exposure equals the source only policy over this horizon. Complete knowledge of the graph therefore supplies an opportunity for effective correction, not a performance guarantee. If all downstream serving endpoints check source acceptance immediately, even a source update can produce zero further stale service; that stronger baseline already implements the essential dependency aware restriction.

If q is discovered to rest on the same withdrawn evidence as p, its independence was misclassified. The support family for z must then be expanded to the shared primitive origin and reassessed. If q covers only the previous cohort, it cannot support a current cohort answer. Both variations demonstrate that source identity and context certification matter as much as the arithmetic of propagation.

7. Evaluation Design for Future Implementations

Technical and Semantic Validation

A future evaluation should use workflows with a known, independently constructed dependency history. The evaluation collection should contain source versions, generated objects, caches, and logged decision uses. It should include conjunctive support, certified alternatives, scoped exceptions, unsupported additions, and deliberately omitted edges. Reviewers should establish which uses require reassessment before testing a correction procedure. A procedure must not define its own candidate list and then use that same list as the complete reference standard.

Technical validation should measure candidate recall and precision against that independent history, eligibility decisions against adjudicated support, and response behavior during concurrent correction events. A request crossing the epoch boundary should be checked for stale return. Duplicate events, failed workers, offline endpoints, and delayed index refresh should be introduced as failure conditions. Each test should report the capture boundary and identify channels, such as private exports, that the implementation cannot observe.

The comparison should include stronger baselines than simple document replacement. Relevant baselines include an authoritative source resolver at every request, a finite cache lifetime, periodic regeneration, and a blanket block of an affected collection. Those baselines may provide similar protection at lower implementation cost. The incremental question is whether explicit support and targeted propagation reduce missed affected uses or unnecessary disruption enough to justify their capture and review burden.

Semantic validation is distinct. Independent reviewers should assess whether a claimed support bundle actually warrants the statement in its stated context. Review should include plausible but overbroad paraphrases and cases in which a cited source contradicts a qualification omitted from the generated answer. Agreement rates and adjudication procedures should be reported. Automated

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

entailment tools may help prioritize review, but agreement with the same model that generated the answer is not independent validation.

Evaluation Measures and Reporting

At minimum, report notification coverage, disposition completion, time to effective suspension, residual stale exposure, and the rate of incorrectly blocked supported objects. Availability costs should include the time during which valid guidance cannot be accessed. Past decision review requires separate measures, including how many eligible decisions were reopened and whether affected people obtained the stated remedy. A short time to notification cannot substitute for these outcomes.

The NIST generative AI risk profile supplies a broader governance context for evaluating information integrity and system risks [22]. Model cards and datasheets provide precedents for documenting intended use and limitations [23,24]. A correction report can extend that practice by documenting the boundary, support assumptions, serving enforcement, and unresolved consequences. This report is an application specific proposal, rather than an existing requirement attributed to those sources.

Organizational evaluation should test whether owners interpret notices accurately and complete justified dispositions. Comparable cases, controlled task order, and independent assessment can measure accuracy, review time, escalation, and notification burden. Capture effort, storage, certification costs, and endpoint integration also require measurement. Any participant study would need appropriate consent and oversight; none was conducted here. Priority policies should disclose whose interests determine consequence weights and how low visibility cases receive attention.

8. Discussion

Human Authority Disagreement and Contestability

An institution must determine who can challenge a warrant, who can temporarily restrict its use, and who can restore service. These powers should be recorded separately. A person affected by an answer may possess decisive contextual information without being the document's custodian. Restricting correction intake to existing owners can prevent that information from reaching the review process. Conversely, automatically adopting every challenge would make knowledge systems vulnerable to disruption and manipulation.

The appropriate response is an accountable procedure for disagreement. A challenge record should preserve the proposed correction, evidence, affected context, reviewer, reasons for disposition, and route for further review. A rejected challenge can remain discoverable to authorized auditors without making the disputed claim publicly visible. This design treats contestation as part of maintaining knowledge while preserving a distinction between raising a concern and establishing that it is correct.

Fricker's analysis of epistemic injustice concerns how social power can undermine people in their capacity as knowers [25]. Applying that concern here suggests evaluating whose correction reports receive attention and whose explanations are dismissed. This is an interpretive extension to system design. It does not follow from graph completeness: a technically complete record can faithfully preserve a process that gives some people inadequate opportunity to contest the record.

Trust and reliance also require care. Lee and See examine appropriate reliance on automation in relation to context and system characteristics [26]. A correction interface should consequently provide reasons for changed eligibility and the scope of any restriction. Repeated generic warnings can increase alert burden without improving judgment. The proposed protocol is intended to make reliance more assessable; whether it does so requires user evaluation rather than an assumption that additional notices necessarily improve trust.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

Transfer to Other Knowledge Settings

The framework is applicable where organizations can identify and control persistent knowledge objects. Examples include internal policy assistants, technical support collections, and research administration guidance. Transfer requires a local definition of primitive warrants, context, semantic certification, and effective suspension. A small organization may implement these functions through a versioned register and controlled publishing process; a larger one may require automated dependency capture and endpoint acknowledgments.

Research synthesis presents a related application. A corrected article may affect a summary without defeating every conclusion in that summary. Independent evidence may preserve one claim while another requires revision. The support family captures this distinction only if the reviewer has recorded which evidence supports which claim. A bibliography at the end of a synthesis cannot provide that information by itself.

The boundary becomes weaker when content enters personal notes, public repositories, or a model's training data. Those channels can be included as known exports with notification duties, but their subsequent use may be unobservable. Institutions should report that limitation and provide durable correction notices at controlled sources. They cannot infer that a public correction has reached every later user.

9. Limitations and Alternative Interpretations

The first limitation concerns the representation of semantic dependence. A language model may combine explicit passages with unstated background assumptions, and humans may copy only part of an answer. The graph cannot capture what the workflow never records. A conservative capture policy can record all retrieved inputs, but doing so can produce many candidate dependencies that were not material to the claim. Precision and recall therefore depend on the capture method, and the formal reachability result should not be mistaken for a solution to provenance discovery.

Second, the positive support language is intentionally restrictive. Real policies contain exceptions, priority rules, default assumptions, and conflicts between authorities. Such reasoning may require a richer truth maintenance or argumentation system. The current model can represent reviewed conclusions from those processes as versioned warrants, but that moves some reasoning outside its formal boundary. A future extension should define conflict resolution explicitly before attempting stronger eligibility guarantees.

Third, certification is a point of possible failure. A reviewer can misunderstand a policy or approve a support family too broadly. An automated checker can reproduce the generator's error. Recording a certificate makes the judgment inspectable, but does not validate it. The design needs independent review sampling, clear scope limits, and procedures for correcting certificates themselves. Otherwise, a polished provenance history can confer unwarranted confidence on a weak inference.

Fourth, the worked example compares stipulated policies and times. It cannot estimate implementation benefits, error prevalence, notification compliance, or the organizational cost of suspension. A periodic refresh policy could outperform targeted propagation in a small, low consequence collection. A source resolver at every request could make persistent derived answers unnecessary. The protocol is most relevant when reuse across multiple persistent artifacts creates dependencies that those simpler arrangements do not already control.

Fifth, exposure weights are contestable. High frequency use need not correspond to the greatest harm, and some harms cannot be aggregated responsibly. The example's units provide an arithmetic illustration. Operational evaluation should disclose alternative weights, report unweighted outcomes, and preserve separate treatment of severe cases. It should also assess burdens created by delay or lost access, which the stale exposure measure excludes.

Sixth, a correction can be politically or organizationally disputed. An approved source may be inaccurate, discriminatory, or inappropriate for a particular person. The acceptance predicate

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

represents the institution's decision to authorize a warrant, not an endorsement of that decision. Contestability and independent scrutiny remain necessary even when every technical condition for eligibility is satisfied.

Finally, the literature selection is focused and purposive. It establishes relevant precedents and supports a bounded synthesis; it does not demonstrate that the proposed combination is absent from all existing commercial systems or research prototypes. The contribution should be assessed as a transparent theoretical specification with testable implications. Claims of empirical effectiveness, comprehensive novelty, or suitability across all sectors would require additional evidence.

10. Conclusions

Correcting a source and completing a correction across organizational knowledge are different operations. The proposed framework makes the relationship explicit through immutable versions, contextual support bundles, semantic certification, and current service eligibility. Its conditional results show what complete dependency records permit, why withdrawing support cannot by itself create eligibility in the restricted model, and why incomplete records limit precise notification guarantees.

The practical consequence is a need to track separate obligations. An organization should establish which dependent objects were examined, whether they retain accepted support, when unsupported service stopped, and which previous decisions require review. A surviving alternative warrant can justify retaining a scoped claim. A delivered notification cannot establish that an obsolete cache has stopped serving or that an affected person received a remedy. Separate dispositions help locate whether a remaining shortfall concerns capture, enforcement, interpretation, or the authority to act.

Future work should evaluate the specification against independently documented workflows and strong maintenance baselines. The decisive outcomes are missed affected uses, inappropriate blocking of supported guidance, stale service during concurrent updates, review burden, and the treatment of past consequences. A credible implementation will need both technical enforcement and people with sufficient information and power to act on a correction.

Author Contributions: C.N. is the sole author and is responsible for conceptualization, methodology, formal analysis, writing, and the final manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable. This theoretical study did not involve human participants, animals, or identifiable personal data.

Informed Consent Statement: Not applicable.

Data Availability Statement: No empirical datasets were collected or analyzed. All constructed objects, dependency relationships, support assumptions, weights, durations, and calculations needed to reproduce the worked example are specified in Section 6 and Table 4.

Acknowledgments: AI tool use: During the preparation of this work, the authors used large language model-based tools to assist with language refinement, text organization, and editorial consistency. After using these tools, the authors reviewed, edited, and validated all generated content to ensure its accuracy, originality, and alignment with the study's scientific contributions. The authors take full responsibility for the content of this publication.

Conflicts of Interest: The author declares no conflicts of interest.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

References

1. Alavi, M.; Leidner, D.E. Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. MIS Quarterly 2001, 25, 107-136. https://doi.org/10.2307/3250961.
2. Nonaka, I. A Dynamic Theory of Organizational Knowledge Creation. Organization Science 1994, 5, 14-37. https://doi.org/10.1287/orsc.5.1.14.
3. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems 2020, 33, 9459-9474. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.
4. Gao, T.; Yen, H.; Yu, J.; Chen, D. Enabling Large Language Models to Generate Text with Citations. In Proceedings of EMNLP 2023; Association for Computational Linguistics: Singapore, 2023; pp. 6465-6488. https://doi.org/10.18653/v1/2023.emnlp-main.398.
5. Buneman, P.; Khanna, S.; Tan, W.-C. Why and Where: A Characterization of Data Provenance. In Database Theory ICDT 2001; Lecture Notes in Computer Science, Volume 1973; Springer: Berlin, Heidelberg, 2001; pp. 316-330. https://doi.org/10.1007/3-540-44503-X_20.
6. Green, T.J.; Karvounarakis, G.; Tannen, V. Provenance Semirings. In Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems; ACM: New York, NY, USA, 2007; pp. 31-40. https://doi.org/10.1145/1265530.1265535.
7. Doyle, J. A Truth Maintenance System. Artificial Intelligence 1979, 12, 231-272. https://doi.org/10.1016/0004-3702(79)90008-0.
8. de Kleer, J. An Assumption-Based TMS. Artificial Intelligence 1986, 28, 127-162. https://doi.org/10.1016/0004-3702(86)90080-9.
9. Moreau, L.; Missier, P., Eds. PROV-DM: The PROV Data Model. W3C Recommendation, 30 April 2013. Available online: https://www.w3.org/TR/prov-dm/ (accessed on 20 October 2026).
10. Cheney, J.; Chiticariu, L.; Tan, W.-C. Provenance in Databases: Why, How, and Where. Foundations and Trends in Databases 2009, 1, 379-474. https://doi.org/10.1561/1900000006.
11. Herschel, M.; Diestelkämper, R.; Ben Lahmar, H. A Survey on Provenance: What For? What Form? What From? The VLDB Journal 2017, 26, 881-906. https://doi.org/10.1007/s00778-017-0486-1.
12. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.J.; Madotto, A.; Fung, P. Survey of Hallucination in Natural Language Generation. ACM Computing Surveys 2023, 55, Article 248. https://doi.org/10.1145/3571730.
13. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In Proceedings of the International Conference on Learning Representations, 2024. https://proceedings.iclr.cc/paper_files/paper/2024/hash/25f7be9694d7b32d5cc670927b8091e1-Abstract-Conference.html.
14. Yan, S.-Q.; Gu, J.-C.; Zhu, Y.; Ling, Z.-H. Corrective Retrieval Augmented Generation. arXiv 2024, arXiv:2401.15884, version 3. https://doi.org/10.48550/arXiv.2401.15884.
15. Hwang, J.; Park, J.; Park, H.; Kim, D.; Park, S.; Ok, J. Retrieval-Augmented Generation with Estimation of Source Reliability. In Proceedings of EMNLP 2025; Association for Computational Linguistics: Suzhou, China, 2025; pp. 34279-34303. https://doi.org/10.18653/v1/2025.emnlp-main.1738.
16. Zhong, Z.; Wu, Z.; Manning, C.; Potts, C.; Chen, D. MQuAKE: Assessing Knowledge Editing in Language Models via Multi-Hop Questions. In Proceedings of EMNLP 2023; Association for Computational Linguistics: Singapore, 2023; pp. 15686-15702. https://doi.org/10.18653/v1/2023.emnlp-main.971.
17. Meng, K.; Bau, D.; Andonian, A.; Belinkov, Y. Locating and Editing Factual Associations in GPT. Advances in Neural Information Processing Systems 2022, 35, 17359-17372. https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.
18. Amershi, S.; Weld, D.; Vorvoreanu, M.; Fourney, A.; Nushi, B.; Collisson, P.; et al. Guidelines for Human-AI Interaction. In Proceedings of CHI 2019; ACM: New York, NY, USA, 2019; Paper 3, pp. 1-13. https://doi.org/10.1145/3290605.3300233.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 21 October 2026 https://doi.org/10.20944/preprints202610.2034.v1

19. Raji, I.D.; Smart, A.; White, R.N.; Mitchell, M.; Gebru, T.; Hutchinson, B.; et al. Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing. In Proceedings of FAT* 2020; ACM: New York, NY, USA, 2020; pp. 33-44. https://doi.org/10.1145/3351095.3372873.
20. Sambasivan, N.; Kapania, S.; Highfill, H.; Akrong, D.; Paritosh, P.; Aroyo, L. Everyone Wants to Do the Model Work, Not the Data Work: Data Cascades in High-Stakes AI. In Proceedings of CHI 2021; ACM: New York, NY, USA, 2021; Article 39, pp. 1-15. https://doi.org/10.1145/3411764.3445510.
21. Nitchals, C. Meaningful Human Authority in AI Mediated Decision Making an Authority Vector and Bottleneck Principle. AI and Ethics 2026, 6, Article 552. https://doi.org/10.1007/s43681-026-01414-7.
22. National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile; NIST AI 600-1; NIST: Gaithersburg, MD, USA, 2024. https://doi.org/10.6028/NIST.AI.600-1.
23. Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Spitzer, E.; Raji, I.D.; Gebru, T. Model Cards for Model Reporting. In Proceedings of FAT* 2019; ACM: New York, NY, USA, 2019; pp. 220-229. https://doi.org/10.1145/3287560.3287596.
24. Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J.W.; Wallach, H.; Daumé III, H.; Crawford, K. Datasheets for Datasets. Communications of the ACM 2021, 64, 86-92. https://doi.org/10.1145/3458723.
25. Fricker, M. Epistemic Injustice: Power and the Ethics of Knowing; Oxford University Press: Oxford, UK, 2007. https://doi.org/10.1093/acprof:oso/9780198237907.001.0001.
26. Lee, J.D.; See, K.A. Trust in Automation: Designing for Appropriate Reliance. Human Factors 2004, 46, 50-80. https://doi.org/10.1518/hfes.46.1.50_30392.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.