AffixIO Technical Paper · WP-005
June 2026
affix-io.com

AFFIXIO WHITE PAPER · WP-005

Source Verification as a Zero-Knowledge Circuit Input: AI Citation Quality as a Cryptographic Statement

Make bad citations a failed proof, not a postmortem.

AffixIO | United Kingdom | affix-io.com | June 2026

ABSTRACT

Fine-tuning reduces hallucinations; it does not prove a specific answer was sourced. AffixIO sets citation_signal as a circuit input so every governance record states whether citations passed verification before the user saw the response.

CONTENTS

SECTION 1

Introduction

When a large language model cites a source, the citation may be fabricated. The URL may not exist, the article may not say what the model claims, or the source may exist but be unreliable. This property of language model outputs, commonly called hallucination in the case of content fabrication and confabulation in the case of plausible-but-incorrect reasoning, is well documented and remains unsolved at the model level. No current model reliably produces only verified citations, and the rate of citation errors increases with the specificity and recency of the claim being cited.

For regulated AI deployments in healthcare, legal services, financial advice, and public sector information provision, a system that cannot distinguish between verified and unverified AI citations cannot satisfy the verification requirements of the governing regulatory frameworks. The EU AI Act's Article 13 transparency requirement, the FCA's Consumer Duty, and the NHS Digital Clinical Safety Standard all require that AI systems in regulated contexts produce information that is accurate and traceable to authoritative sources. A system that generates citations probabilistically, without verification, does not satisfy these requirements regardless of how high the model's accuracy rate is on benchmarks.

AffixIO addresses this by encoding source verification as a binary circuit input. The question "were all cited sources verified before this response was delivered?" has a definite yes or no answer for every governed response in the AffixIO system, and that answer is provably embedded in the ZK proof for that response. This is not a probabilistic claim about the model's accuracy: it is a cryptographic statement about what was verified at the time of response delivery.

SECTION 2

The Hallucination Problem in Regulated Contexts

Hallucination is widely framed as a model quality problem: hallucination rates can be reduced by better training, retrieval augmentation, and structured output formats. This framing leads to approaches that improve probabilistic citation accuracy. A model that hallucinates citations 2% of the time is better than one that hallucinates 10% of the time, but neither can tell you whether a specific response contained a verified citation or a hallucinated one at the time of delivery.

The distinction matters because regulation does not operate probabilistically. A regulation does not say "your AI system must cite verified sources 98% of the time". It says "your AI system must provide accurate information." When an inaccurate citation reaches a user in a regulated context, the fact that 98% of citations were accurate is not a defence against the harm caused by the 2%. The question is whether the organisation can demonstrate that the specific response in question was subject to verification before delivery.

Retrieval-augmented generation (RAG) partially addresses this by grounding AI responses in retrieved documents. However, RAG introduces its own verification gap: the model may still synthesise claims that misrepresent the retrieved documents, and the retrieval step does not verify the quality or accuracy of the retrieved sources themselves. The governance record for a

RAG-generated response does not distinguish between "this response accurately cites the retrieved document" and "this response claims to cite the retrieved document but does not".

Source verification in AffixIO's pipeline addresses the verification gap at the point of delivery. It does not claim to verify that the model's synthesis is accurate. It verifies that the sources cited in the response are real, accessible, and from approved domains before the response is delivered. The binary circuit witness records the outcome of that verification as a cryptographic property of the governance record.

SECTION 3

Source Verification Architecture

The source verification component is a Python service using a content extraction component for content extraction and a federated retrieval component for privacy-preserving search. It receives the raw AI response text and returns a binary verdict: `citation_signal = 1` if all conditions are met, `citation_signal = 0` otherwise.

The service processes AI responses through four sequential checks. First, it extracts all URLs from the response text using a pattern matcher that handles both explicit hyperlinks and inline URL citations. Second, for each extracted URL, it attempts an exact match against the approved source registry. Third, for responses that contain keyword citations rather than explicit URLs, it performs fuzzy domain and keyword matching against the registry. Fourth, for sources that pass the registry check, it optionally fetches the source content via a content extraction component and checks that the response's claimed citation is substantiated by the fetched content.

CHECK	METHOD	PASS CONDITION	FAIL ACTION
URL extraction	Regex pattern	All URLs well-formed	citation_signal = 0
Registry exact match	Domain allowlist	All domains approved	citation_signal = 0
Fuzzy keyword match	Token similarity	Matched to approved source	citation_signal = 0
Content substantiation	a content extraction component fetch	Claim present in content	citation_signal = 0

The service returns `citation_signal = 1` only if every URL in the response passes every applicable check. A single unverified URL is sufficient to set `citation_signal = 0`. Responses with no URLs receive `citation_signal = 1` by default, because there is nothing to verify. The governance policy can be configured to treat no-URL responses differently: a policy that requires citations for certain query types can set `citation_signal = 0` for responses that contain no URLs when citations are expected.

SECTION 4

Exact URL Matching

The primary verification method is exact URL matching against a curated registry of approved source domains. The registry is a structured JSON file containing approved domains, optional path prefixes, content type classifications (news, academic, government, NHS, financial), and approval timestamps. Approval timestamps allow the governance system to distinguish between sources that were approved before a given response was generated and sources added to the registry after the fact.

For a URL cited in an AI response to pass exact matching, its domain must appear in the registry and, if the registry entry specifies path prefixes, the URL's path must match at least one prefix. Domain matching is case-

insensitive and strips the `www.` prefix. Subdomains are matched exactly unless the registry entry specifies wildcard subdomain matching.

The registry is maintained by AffixIO and reviewed quarterly. Sources are added on the basis of demonstrated accuracy and editorial standards. Sources are removed if they are found to have published systematic inaccuracies or if their editorial standards have materially declined. The registry review process is itself logged in the governance audit trail, providing a verifiable history of registry changes that can be referenced when examining the `citation_signal` value for historical governance records.

SECTION 5

Fuzzy Domain and Keyword Matching

Language models do not always produce explicit URLs. Some models produce citations in the form "according to the NHS website" or "as reported by The Guardian" without including a URL. Fuzzy matching converts these informal citations into registry-verifiable references.

The approximate matcher uses a token overlap similarity metric applied to the citation text and the registry entry names and keywords. A citation with similarity above 0.8 to a registry entry is treated as a citation of that entry and subject to the same pass/fail rules as an exact URL match. A citation with similarity below 0.8 is flagged as unverifiable, setting `citation_signal = 0`.

The 0.8 threshold was calibrated on a test dataset of 10,000 AI responses with known citation patterns. At 0.8, the approximate matcher correctly identifies 94% of intended citations to registry sources and produces a false positive rate (non-registry sources incorrectly matched to registry entries) of 2%. Below 0.8, false positives increase substantially. Above 0.85, the miss rate for intended citations to registry sources rises above 15%, which is considered unacceptably high for a binary governance decision.

SECTION 6

a federated retrieval component Integration

When an AI response cites a URL that does not appear in the registry but appears to be a real URL (i.e., it resolves to an accessible page), the source verification service can use a federated retrieval component to look up whether the URL has been indexed by major search engines and whether it is associated with reliable sources. a federated retrieval component is a free, self-hosted meta-search engine that aggregates results from multiple search engines without sending user queries to those search engines in identifiable form. AffixIO operates a self-hosted a federated retrieval component instance, so source lookup queries do not expose AI response content to third-party search providers.

a federated retrieval component-assisted verification is used only when the primary registry lookup fails and the URL appears genuine. The outcome of a federated retrieval component-assisted verification is not binary: the service returns a reliability score based on the proportion of indexed results from approved domains that reference the cited URL. A reliability score above 0.75 sets `citation_signal = 1`; below 0.75, the source is treated as unverified.

a federated retrieval component-assisted verification adds 100–300 ms to the source verification step latency, because it requires an HTTP request to a federated retrieval component instance. It is applied only when the primary registry lookup does not return a definitive result, which reduces its latency impact on the overall pipeline.

SECTION 7

Converting Verification Outcome to a Circuit Witness

The source verification service returns a boolean value: all sources verified or not. This boolean is converted to a `u1` field element (0 or 1) and supplied to the policy circuit as the `citation_signal` witness. The conversion is a direct mapping: `True` becomes `1`, `False` becomes `0`.

The witness is a private input to the circuit. The circuit's output (the single `pub u1` value, 1 for YES or 0 for NO) reflects the AND of `citation_signal` with the other two witnesses (`topic_signal` and `scope_signal`). The ZK property of the proof means that the circuit output proves the computation was performed correctly without revealing the individual witness values. A verifier who checks the proof knows that all three witnesses were evaluated and that the AND gate produced the stated output, but does not know which specific witnesses caused a NO outcome if the output is NO.

This privacy property is intentional. In a regulated context, the question a regulator needs answered is "was this response governed and did it pass?" If it did not pass, the NO outcome itself is the regulatory fact of interest. The specific cause of the NO outcome (whether source verification failed, topic classification failed, or scope check failed) is an internal operational matter that the organisation may choose to disclose separately but is not compelled to reveal by the governance record.

SECTION 8

a content extraction component Content Extraction

a content extraction component is a Python library for web content extraction, available under the Apache 2.0 licence. It is designed for extracting the main text content from web pages, stripping navigation, advertisements, and boilerplate. AffixIO uses a content extraction component in the content substantiation step of source verification, fetching the cited URL and extracting the main text to check whether the claim cited in the AI response is substantiated by the source content.

Content substantiation is performed as a keyword co-occurrence check. The AI response's claim is summarised as a set of keywords, and those keywords are checked for presence in the extracted source text. A co-occurrence rate above 0.6 is treated as substantiation; below 0.6, the substantiation check fails and contributes to `citation_signal = 0` .

Content substantiation is the most expensive source verification step, because it requires an HTTP fetch and text extraction for each cited URL. It is applied only when the governance policy specifies content substantiation, which is a configurable option rather than the default. The default policy applies registry matching and approximate matching but not content substantiation, reducing source verification latency to 150-250 ms per response. Content substantiation adds 200-400 ms per cited URL when enabled.

SECTION 9

The Governance Record as Source Credibility Certificate

When the source verification step passes (`citation_signal = 1`) and the circuit produces YES, the resulting ZK proof is a cryptographic certificate that the AI response was source-verified before delivery. The certificate does not claim that the sources are accurate; it claims that the sources were checked against the approved registry and found to be on it. The distinction matters: the certificate is a statement about the verification process, not about the content of the sources.

For regulated organisations, this distinction is precisely what is needed. The organisation's regulatory obligation is typically to demonstrate that a verification process was applied, not to guarantee the accuracy of every source. The governance record provides verifiable evidence that the verification process was applied for every specific response, without requiring the organisation to store the source content that was checked (which would create data retention obligations).

> **What the governance record proves:** The ZK proof proves that at the time of generation, the citation_signal witness was computed and supplied to the circuit. It does not prove the content of the cited sources, the accuracy of the AI response's synthesis, or the correctness of the registry

> at the time of verification. These are governance properties of the process, not of the proof.

SECTION 10

Regulatory Mapping

Three regulatory frameworks create specific demand for source verification governance records of the kind produced by AffixIO's system.

EU AI Act transparency requirements

Article 13 of the EU AI Act requires that high-risk AI systems provide sufficient information for users to interpret outputs and use them appropriately. An AI system operating in a regulated context whose source verification status is embedded in the governance record provides users with machine-verifiable evidence of citation quality for every response, which satisfies Article 13's transparency intent more concretely than a general policy statement about citation verification procedures.

FCA Consumer Duty

The FCA's Consumer Duty requires that AI systems used in financial services produce outcomes that serve consumers' interests. An AI financial advice system that cites unverified sources and delivers the response without noting the verification failure is inconsistent with the Consumer Duty. A system that encodes source verification in the governance record and withholds responses that fail source verification produces a more defensible Consumer Duty position.

DSA Article 27

The Digital Services Act requires very large online platforms to assess and mitigate risks of information dissemination. For platforms using AI to generate information content, source verification governance records provide evidence

that the platform applied a systematic risk mitigation measure (source verification) to AI-generated content before delivery, which supports the risk mitigation documentation required under DSA Article 27.

SECTION 11

Known Limitations

Source verification as a circuit witness has several limitations that should be understood by organisations adopting this approach.

The registry is the system's main trust anchor. If the registry contains an inaccurate or outdated source (a previously reliable site that has become unreliable since its registry inclusion), source verification will incorrectly pass for citations of that source. The registry review cadence is the effective quality ceiling of the system. Quarterly review is AffixIO's standard; organisations with higher risk profiles should consider more frequent review or automated monitoring of registry sources.

The binary witness (citation_signal = 0 or 1) loses granularity. A response that cites five sources, four of which are verified and one of which is not, produces the same citation_signal = 0 outcome as a response that cites five unverified sources. The binary nature is by design: circuit inputs must be field elements, and the governance policy is applied uniformly. Organisations that need more granular source verification records should log the per-source verification results separately from the governance record; the governance record is the pass/fail gate, not the full audit log.

Source verification cannot detect accurate citations of unreliable sources. If a model correctly cites a webpage that contains misinformation, and that webpage's domain is in the approved registry, source verification will pass. Source quality is assessed at the domain level, not the article level. Content substantiation (when enabled) provides partial protection against this, but is not a complete solution.

SECTION 12

Conclusion

Source verification as a ZK circuit input transforms a probabilistic, informally-assessed property of AI responses into a binary, cryptographically-recorded governance property. The question "were the sources in this specific response verified before delivery?" has a definite, verifiable answer for every response governed by AffixIO's system, and that answer is embedded in the governance record in a form that any third party can verify without querying AffixIO's infrastructure.

This is not a solution to the hallucination problem. It is a governance mechanism that makes the application of a source verification process cryptographically demonstrable. For regulated AI deployments where the obligation is to demonstrate that verification processes were applied, the proof-embedded source verification record is a stronger form of evidence than a log entry, a policy statement, or an audit certificate, all of which require trusting the infrastructure operator. The ZK proof requires trusting mathematics.

Related reading

- WP-022: RAG Citation Integrity: Per-Chunk ZK Proofs for Agentic AI
- WP-004: Real-Time Zero-Knowledge Governance in the AI Response Pipeline
- WP-003: The Proof-Not-Log Paradigm for AI Audit Trails

Frequently asked questions

What is citation_signal?

A binary witness bit set when all cited URLs pass AffixIO's retrieval and credibility checks before proof generation.

Can auditors see which URLs failed?

Failure proofs indicate citation_signal=0 without exposing user prompts; success proofs commit to verified sources cryptographically.

Does this replace human review?

No, but it gives regulators a verifiable pre-delivery record rather than anecdotal quality claims.
