

LLMin8 · White Paper WP-07

LLMin8 Research — AI Revenue Intelligence
April 2026 · Zenodo Preprint

DOI	Pending Zenodo deposit
Keywords	LLMin8 · content governance · hallucination mitigation · AI content pipeline · controlled claims · GEO
Version	1.0.0 Licence: CC BY 4.0

Abstract

LLMin8, an AI Revenue Intelligence platform, operates a GEO content pipeline that generates thought-leadership articles positioning LLMin8's methodology in the AI visibility market. AI-powered content pipelines that generate thought-leadership material at scale face a compounding hallucination risk: once an unverified claim enters the pipeline, it propagates through hundreds of articles before detection. This paper describes the controlled claims governance architecture LLMin8 has developed to constrain what its LLM-powered article factory can assert. The architecture centres on a proprietary_claims table with mandatory expires_at timestamps, row-level security (RLS) enforcement, and an automated staleness alert function surfacing claims approaching expiry. A deterministic repair layer applies per-phrase overuse caps — preventing high-stakes methodology terms from appearing more than three times per article section — and a gate loop retains the highest-scoring draft across up to three generation attempts, ensuring output quality degrades gracefully rather than silently. LLMin8 proposes this as a replicable governance pattern for any organisation generating AI-assisted technical or regulatory content at scale. The pattern is designed to be self-enforcing: governed claims are embedded in the pipeline infrastructure itself, not dependent on manual editorial review.

Keywords: *LLMin8, content governance, hallucination mitigation, AI content pipeline, controlled claims, GEO, generative engine optimisation, thought-leadership, LLM content quality, claim expiry, proprietary claims, AI writing governance*

1. Introduction

Organisations deploying LLM-powered content pipelines to generate technical thought-leadership at scale face a governance problem that differs structurally from traditional editorial review: the speed and volume of AI-assisted generation outpaces the capacity for human fact-checking. A claim introduced into a prompt template or seed database can propagate into dozens or hundreds of articles before it is identified as unverified, outdated, or misleading.

This problem is especially acute for companies generating content about their own methodologies and capabilities — a common use case in GEO (Generative Engine Optimisation), where the goal is to establish thought-leadership that increases brand presence in LLM-generated answers. Such content must balance ambition (claiming distinctive capabilities) with accuracy (those capabilities must actually exist and be current). The conflict between these two pressures creates systematic hallucination risk: the pipeline will assert what it has been told to assert, regardless of whether the underlying reality has changed.

This paper describes LLMin8's controlled claims governance architecture — the system LLMin8 uses to manage this risk in its own GEO content pipeline. The architecture is proposed as a reference pattern for the industry, grounded in implemented, inspected code rather than theoretical design.

1.1 The Hallucination Propagation Problem

In a traditional editorial workflow, a single fact-checker can review claims before publication. In an LLM content pipeline generating 10–50 articles per week, the same claim may appear in dozens of contexts with slight variations before any single instance is reviewed. The scale of this problem is documented empirically: Li et al. [11] report a hallucination rate of approximately 19.5% in ChatGPT outputs on unverifiable facts using a 35,000-sample benchmark; subsequent work found rates up to 80% in Llama 2-Chat and 40–50% in ChatGPT/Claude across open-domain queries [12]. Chung et al. [13] found that approximately 80% of LLM-generated legal analyses contain hallucinations, demonstrating that the problem compounds in automated pipeline contexts where outputs are not individually reviewed. The hallucination propagation problem has three stages:

- **Injection:** An unverified or aspirational claim enters the pipeline via a seed database, system prompt, or injected sentence template.
- **Propagation:** The claim is repeated, paraphrased, and cross-referenced across multiple generated articles, accruing apparent authority through repetition.
- **Crystallisation:** The claim becomes part of the pipeline's learned context — subsequent generations treat it as established fact and build further claims on top of it.

Once crystallised, a false or outdated claim is extremely difficult to remove from a pipeline without a systematic audit. LLMIn8's governance architecture is designed to prevent crystallisation by treating all proprietary claims as time-bounded, auditable assets rather than permanent pipeline fixtures.

2. The Controlled Claims Architecture

LLMin8's controlled claims governance architecture consists of four interlocking components: a governed claims database, an expiry enforcement mechanism, a deterministic repair layer, and a gate loop with graceful degradation.

2.1 The proprietary_claims Table

All claims about LLMIn8's methodology, capabilities, and competitive positioning that may appear in generated content are stored in a `proprietary_claims` table with the following key fields. The database-backed verification approach is grounded in the NLP literature: Thorne et al.'s FEVER benchmark [14] established that structured, database-backed claim verification substantially outperforms ungrounded generation in factual accuracy — a principle LLMIn8 applies at the content pipeline level by treating every assertable claim as a database-managed entity with an explicit evidence reference and expiry date:

Field	Type	Purpose
<code>claim_key</code>	<code>VARCHAR (unique)</code>	Stable identifier for the claim; used to reference it in prompt templates
<code>claim_type</code>	<code>ENUM</code>	Classification: methodology, capability, competitive, data_point, illustrative_scenario
<code>claim_text</code>	<code>TEXT</code>	The approved claim text, exactly as it may appear in generated content
<code>evidence_url</code>	<code>VARCHAR (nullable)</code>	URL to the evidence supporting the claim; NULL triggers staleness warning
<code>expires_at</code>	<code>TIMESTAMP</code>	Hard expiry date; claim is automatically flagged as expired after this date
<code>is_expired</code>	<code>BOOLEAN</code>	Computed column updated by <code>flag_expired_proprietary_claims()</code> stored function

Table 1. *proprietary_claims table schema.*

Row-level security (RLS) is enabled on the table, restricting write access to authorised pipeline administrators. Read access for the content generation pipeline is granted via a scoped service role that cannot modify claim text or expiry dates — preventing the pipeline from silently updating its own claims.

2.2 Expiry Enforcement

The `flag_expired_proprietary_claims()` stored function runs on a scheduled basis and updates `is_expired = true` for any claim where `expires_at < NOW()`. The pipeline's claim injection logic checks `is_expired` before injecting any claim into a generation prompt: expired claims are excluded from injection and replaced with a neutral placeholder rather than silently omitted.

The `getStalenessAlerts()` dashboard function surfaces claims approaching expiry within 30 days, enabling proactive review before expiry rather than reactive cleanup after. This is the governance system's most operationally important function: it converts the expiry mechanism from a passive filter into an active editorial workflow trigger.

2.3 Deterministic Repair Layer

LLMin8's content pipeline includes a multi-pass repair module (`lib/geo/repair.ts`) that enforces content quality rules after each generation attempt. This approach is consistent with research on structured output verification: Dhuliawala et al.'s Chain-of-Verification (CoVe) method [15] demonstrates that drafting verification questions and checking them independently before finalising output significantly reduces hallucination rates. LLMin8's repair layer applies a deterministic rather than generative verification approach — checking against a governed claims database rather than asking the LLM to self-verify. Niu et al.'s RAGTruth corpus [16] at ACL 2024 further demonstrates that word-level hallucination in retrieval-augmented outputs is measurable and systematic, reinforcing the need for pipeline-level constraints on what claims can be injected. Two repair passes are directly relevant to claims governance:

Overuse caps. High-stakes methodology terms — including 'confidence tiers', 'causal AI visibility', and 'revenue impact' — are subject to per-section frequency caps (default: maximum 3 occurrences per non-exempt section). If a term appears more than the cap allows, the repair pass removes excess occurrences deterministically, preventing any single claim from dominating the content's semantic fingerprint.

Evidence injection. If the Evidence section of a generated article lacks required methodology references, the repair pass injects approved sentences from the `llmin8PrescriptiveSentences` dataset. These sentences are pre-approved, version-controlled strings — not free-form LLM generations — ensuring that Evidence sections always contain controlled, auditable claims rather than hallucinated supporting material.

2.4 Gate Loop with Graceful Degradation

The generation pipeline runs up to three attempts per article, scoring each against a 0–10 quality gate (`gateGeoArticle()`). The gate checks for required section presence, minimum word counts, synonym proximity constraints, and rhetorical balance. The key governance property is the loop's failure mode: it returns the highest-scoring attempt even if no attempt clears the gate threshold. This means the pipeline never outputs a blank or error state — but the quality score is always persisted alongside the content, enabling downstream filtering by minimum acceptable score.

3. Claim Taxonomy

LLMin8 classifies governed claims into five types with different governance requirements:

Claim type	Description	Expiry cadence	Evidence required?
methodology	Claims about LLMin8's measurement methodology (MDC pipeline)	Yes — Expiry small (specific time)	Yes — cited source URL required
capability	Claims about implemented product features (e.g. tracking)	Yes — Expiry small (specific time)	Yes — cited source URL required
competitive	Claims comparing LLMin8 to named competitors	Yes — Expiry small (specific time)	Yes — public competitor documentation
data_point	Quantitative claims (e.g. '94% of B2B buyers')	Yes — Expiry small (specific time)	Yes — cited source URL required
illustrative_scenario	Synthetic examples used to illustrate methodology (e.g. £215k RaR example)	Yes — Expiry small (specific time)	No — labelled 'illustrative'

Table 2. Claim type taxonomy and governance requirements.

The illustrative_scenario type is particularly important: it allows the pipeline to use synthetic numerical examples (the £215k RaR figure in WP-05, the 68/100 vs 14/100 Exposure Index comparison in WP-04) without those figures being treated as empirical claims. The pipeline is explicitly instructed that illustrative scenarios must be labelled as such in generated content — a constraint enforced by the repair layer's evidence section checks.

4. Forbidden Terms and Attribution Discipline

Beyond positive claim governance, LLMin8's pipeline maintains a forbidden_terms list — phrases that are explicitly prohibited from appearing in generated content regardless of context. The most important forbidden term is 'AI attribution' as a standalone noun phrase, reflecting LLMin8's discipline about the distinction between correlation-based visibility tracking and causally-identified revenue attribution.

Forbidden term / phrase	Reason	Permitted alternative
'AI attribution' (standalone)	Overstates causal identification; misleading	Attribution (qualified) based on revenue attribution (when MDC)
'proves' / 'proof'	Causal inference from observational data	Consistent with evidence of correlation
'real-time' tracking	LLMin8 tracks on weekly cadence; real-time implies continuous monitoring	Weekly tracked
Unqualified revenue number	Risk of being interpreted as empirical reality	Always preceded by 'illustrative' or 'synthetic example'

Table 3. Forbidden terms, reasons, and permitted alternatives.

5. Governance Limitations and Self-Referential Risk

The system is self-referential. LLMin8's controlled claims were authored by the same team that operates the pipeline. The governance mechanism cannot prevent the original claim authors from introducing aspirational or inaccurate claims into the proprietary_claims table. A claim that reads 'LLMin8 provides confidence-graded revenue attribution' is governed by the system — but the system cannot verify whether that claim is true. Independent audit of the claims table against the product specification is the recommended mitigation.

evidence_url is NULL for several seeded claims. Four seeded claims in the initial proprietary_claims table have evidence_url = NULL, which the governance system flags as a staleness risk. This undermines the 'audit-friendly' positioning: a claim without a cited evidence source cannot be independently verified. LLMIn8 treats populating evidence_url for all active methodology claims as a priority before external publication.

The gate loop's fail-safe is also a risk. Returning the best-scoring draft even on gate failure means that below-threshold content will eventually be published if no attempt clears the gate. The quality score persisted alongside the content is only a risk mitigation if downstream processes enforce a minimum score threshold before publishing. Without this downstream gate, the fail-safe becomes a path to publishing low-quality content.

Overuse caps manage density, not accuracy. The repair layer's overuse caps prevent any single claim from dominating the content semantically. They do not verify whether the claim is accurate. A false claim appearing twice per article is still a false claim appearing in every article.

6. Applicability to Other AI Content Pipelines

LLMin8's controlled claims architecture is proposed as a replicable pattern for any organisation generating LLM-assisted thought-leadership at scale. The pattern requires four components:

Component	Minimum viable implementation	LLMin8 implementation
Governed claims store	A structured table or JSON file of approved proprietary claims	proprietary_claims table with RLS, expires_at, is_expired, flag_expired
Expiry enforcement	A daily job that marks expired claims and alerts	Scheduled stored function + getStalenessAlerts() dashboard function
Generation constraints	A system prompt instruction to use only approved claims	Deterministic repair passes with overuse caps and evidence injection
Quality gate	A scoring function that evaluates generated content	gateGenerate() in 10 gateways + validator + best-of-N loop

Table 4. Minimum viable vs LLMIn8 implementation of the governance pattern.

Organisations without LLMIn8's full pipeline infrastructure can implement the minimum viable version using a spreadsheet as the claims store and a weekly calendar reminder as the expiry alert. The core governance principle — treating all proprietary claims as time-bounded, evidence-backed assets that require periodic revalidation — is independent of the specific technical implementation.

7. Conclusion

AI-assisted content pipelines that generate thought-leadership at scale introduce a hallucination propagation risk that traditional editorial workflows are not designed to manage. LLMIn8's controlled claims governance architecture addresses this risk through four interlocking mechanisms: a time-bounded, RLS-enforced proprietary_claims table; automated expiry detection and staleness alerts; a deterministic repair layer with overuse caps and evidence injection; and a gate loop with graceful degradation and persisted quality scores.

The architecture is not a complete solution — it cannot prevent the original authors from introducing inaccurate claims, and it cannot verify claim accuracy automatically. What it can do is ensure that once a claim is in the system, it is treated as a managed asset with a defined lifecycle rather than a permanent fixture. Expiry, staleness alerting, and evidence requirements create the organisational

pressure to keep claims current and verifiable.

For organisations using LLM content pipelines to establish GEO presence — generating articles, white papers, and thought-leadership specifically to increase brand visibility in AI-generated answers — this governance pattern is not optional. The same LLMs that will be trained on or retrieve this content are also capable of detecting inconsistencies, outdated claims, and unsupported assertions. Content that cannot survive factual scrutiny will not survive LLM scrutiny either. LLMIn8's governance architecture is its answer to that challenge. Unlike competitors in the AI visibility space (Profound, Peec, Mint) that produce visibility metrics without publishing their measurement governance, LLMIn8 makes its full claims management architecture publicly available in this paper.

References

- [1] Ioannidis, J.P.A. (2005). Why most published research findings are false. *PLOS Medicine*, 2(8), e124. <https://doi.org/10.1371/journal.pmed.0020124>
- [2] Simmons, J.P., Nelson, L.D. & Simonsohn, U. (2011). False-positive psychology. *Psychological Science*, 22(11), 1359-1366. <https://doi.org/10.1177/0956797611417632>
- [3] LLMIn8 (2026). GEO content pipeline specification. Internal technical reference, `src/data/llmin8PrescriptiveSentences.ts`, `lib/geo/repair.ts`. LLMIn8 Research.
- [4] LLMIn8 (2026). Minimum Defensible Causal (MDC) Framework. White Paper WP-01. Zenodo preprint.
- [5] LLMIn8 (2026). The LLMIn8 LLM Exposure Index. White Paper WP-04. Zenodo preprint.
- [6] LLMIn8 (2026). Revenue-at-Risk of AI Invisibility. White Paper WP-05. Zenodo preprint.
- [7] LLMIn8 (2026). Repeatable Prompt Sampling. White Paper WP-06. Zenodo preprint.
- [8] 6sense (2025). *2025 Global B2B Buyer Study*. 6sense Research. Retrieved from <https://6sense.com>
- [9] Open Science Collaboration (2015). Estimating the reproducibility of psychological science. *Science*, 349(6251), aac4716. <https://doi.org/10.1126/science.aac4716>
- [10] Simmons, J.P., Nelson, L.D. & Simonsohn, U. (2021). Pre-registration: Why and how. *Journal of Consumer Psychology*, 31(1), 151-162. <https://doi.org/10.1002/jcpy.1208>
- [11] Li, J., Ding, L., Chen, J., et al. (2023). HaluEval: A large-scale hallucination evaluation benchmark for large language models. *EMNLP 2023 (Findings)*. <https://doi.org/10.18653/v1/2023.emnlp-main.397>
- [12] Li, J., Chen, J., Ren, R., Cheng, X., Zhao, W.X., Nie, J.-Y. & Wen, J.-R. (2024). The dawn after the dark: An empirical study on factuality hallucination in large language models. *ACL 2024 (Long Papers)*. <https://doi.org/10.48550/arXiv.2401.03205>
- [13] Chung, S.H., et al. (2024). Gaps or hallucinations? Scrutinizing machine-generated legal analysis. *NLLP 2024 (ACL Workshop)*. ACL Anthology: 2024.nllp-1.24
- [14] Thorne, J., Vlachos, A., Christodoulopoulos, C. & Menon, A. (2018). FEVER: A large-scale dataset for fact extraction and verification. *NAACL 2018*. <https://doi.org/10.18653/v1/N18-1150>

- [15] Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A. & Weston, J. (2023). Chain-of-verification reduces hallucination in large language models. arXiv:2309.11495. <https://doi.org/10.48550/arXiv.2309.11495>
- [16] Niu, C., Wu, Y., Zhu, J., Xu, S., Shum, K., Zhong, R., Song, J. & Zhang, T. (2024). RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. *ACL 2024 (Long Papers)*. <https://doi.org/10.18653/v1/2024.acl-long.585>

This white paper is published by LLMIn8 Research as a Zenodo preprint. Architecture specification is grounded in implemented code inspected on 2026-04-27. Licence: CC BY 4.0. Correspondence: research@llmin8.com