

Trust Chains for AI-Assisted Heritage Knowledge: A Wiki-Backed, Tradition-Aware Pipeline

Vijendra Kushwaha¹

¹Indian Institute of Technology Kanpur, India

Abstract

AI-assisted knowledge engineering raises a question that semantic technologies are well placed to answer: how does a reader know which assertions to trust, and on whose authority? We demonstrate the Heritage Semantic System (HSS), a deployed platform for Sanskrit classical texts that combines an open MediaWiki, a small pool of trusted scholarly editors, and an LLM translation bootstrap into a knowledge graph in which every assertion — source verse, transliteration, translation, commentary, even a chapter title in one of eleven Indian languages — carries an auditable provenance chain, inspectable through the system’s API today and materialised as OWL/RDF over LRMoo, OntoLex-Lemon, PROV-O, and SKOS. A typed editorial state machine routes contributions by trust level; LLM drafts enter on a dedicated bot track and are never promoted without expert review. Interpretive tradition is a first-class dimension of the schema, so parallel commentary lineages coexist rather than being collapsed into one canonical answer. We demonstrate the system end-to-end on the Bhagavad Gītā and report competency questions answered by the deployed system.

Keywords

human–AI knowledge engineering, provenance, PROV-O, cultural heritage, multilinguality

1. Introduction

When a language model drafts a translation of a sixth-century verse and a reader encounters that translation two years later, what stands between the two is a chain of evidence: who generated the draft, who reviewed it, against which version of the source, and under what editorial authority. If that chain is missing, the reader is asked to trust a black box; if it exists only as prose in an “about” page, it cannot be inspected at the level of the individual assertion. Our position is that this chain should be a graph object — typed, attached to every assertion, and queryable — and that existing semantic standards (PROV-O [1], LRMoo [2], SKOS [3]) are already sufficient to build it.

The Heritage Semantic System (HSS) is a running implementation of this position on a culturally significant corpus. It serves the Bhagavad Gītā (701 verses carrying 8,851 translations, 11,170 commentaries and 2,798 commentary translations) and the Vālmiki Rāmāyaṇa (22,477 verses, 37,378 translation segments, 20,198 commentary segments) through a pipeline in which an open wiki is the source of truth, a governed state machine mediates promotion into the knowledge graph, and LLM-generated drafts are structurally distinguishable from human scholarship at every point in their life cycle.

Two design choices distinguish HSS from related corpus platforms such as GRETEL [5] and the Digital Corpus of Sanskrit [6], and from collaborative knowledge bases such as Wikidata [7]. First, the trust chain itself is part of the graph: each content expression points back through the editor, their trust level, the frozen snapshot that was reviewed, and the state transition that promoted it. Second, interpretive *tradition* (Sanskrit *paramparā* — the commentarial lineage a text has been read through, e.g. Advaita or Viśiṣṭādvaita) is a first-class dimension, so the graph can hold several mutually incompatible readings of the same verse without ranking them. This is a deliberate epistemic stance: for texts with a millennium of contested interpretation, a “single source of truth” design would itself be an editorial act, and the wrong one.

Posters, Demos, Blue Sky, and Tutorials at SEMANTiCS 2026, Sep 2026, Ghent, Belgium

EMAIL: vijendr@iitk.ac.in (V. Kushwaha)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Live and publicly accessible since June 2026 (<https://heritage.iitk.ac.in/>); content seeding and editorial onboarding are ongoing in parallel with this submission.

The closest existing formalism is expressing a claim in a knowledge graph without asserting it — the modelling of *conjectures* [8], whose RDF realisations have since been evaluated [9], developed for exactly the scholarly situation where sources disagree. HSS needs a neighbouring but distinct move. A conjecture is un-asserted, held at arm’s length pending evidence; the readings we hold in parallel are each fully asserted, and each true within the frame of the tradition that holds it. What varies is not the confidence attached to a claim but the interpretive frame relative to which it is evaluated. We therefore keep tradition as a dimension of the assertion rather than a qualifier on its certainty, and reserve lifecycle state (*ai_drafted*, *community_vetted*, *authoritative*) for the orthogonal question of editorial confidence.

2. System Overview

HSS is a three-tier stack: MediaWiki as the canonical content store, a Django service layer holding synced snapshots and structured metadata, and a React reading interface. The graph is exported in OWL/RDF over LRMoo, OntoLex-Lemon [4], PROV-O, and SKOS. The vocabulary is published under <https://heritage.iitk.ac.in/ontology/hss>, which dereferences to Turtle; the *hss:* and *hssr:* prefixes in the figures below resolve there.

Multi-lens verse model. Each verse is modelled as an LRMoo F1 work realised by independently addressable expressions: the Sanskrit source text (*mūlam*), transliterations, translations by named authors, commentaries tagged by tradition, commentary translations, and audio recitations. Rendering the same expression in a different script is a manifestation-level operation, not a new expression.

Tradition as a dimension. Every interpretive artifact — summary, introduction, localised title, annotation, concept definition — carries an indexed *parampara* (tradition) field, and the reader interface resolves content through a fallback cascade (requested language and tradition, then language, then English, then any) that never substitutes one tradition’s content for another’s: a reader who selects Advaita and finds no Advaita summary is shown the gap, not a Dvaita summary presented as neutral. BG 2.47 carries fifteen commentaries side by side, unranked, and each resolves to its lineage in the deployed API: Advaita for Śaṅkara, Ānandagiri and Madhusūdana, Viśiṣṭādvaita for Rāmānuja and Vedānta Deśika, Dvaita for Madhva and Jayatīrtha, Śuddhādvaita for Vallabha, Kashmir Śaiva for Abhinavagupta. A translation inherits the lineage of the commentary it renders rather than of its translator: a rendering of Madhva’s reading stays Dvaita in an Advaitin’s English. Two limits we state rather than hide — the code is *derived* (from the attributed author’s tradition, else a curated commentator registry) rather than stored per row, so lineage-scoped querying is coarser than lineage-scoped display; and coverage is dense for the Gītā, partial elsewhere.

Multilinguality at every layer. Language tags are not confined to the translation lens. Chapter titles, section headings, verse labels, and summaries are all language-tagged entities that travel through the same governance pipeline, so a Tamil reader sees a Tamil-titled chapter containing Tamil-script source text and a Tamil translation, each with its own provenance.

Cross-script substrate. Sanskrit is one language written in many scripts. HSS folds them into a single normal form — script auto-detection, transliteration to IAST, diacritic folding — stored as *content_norm* on each of the 105,121 artifacts in the deployed search index (20,001 for the Gītā alone). The same query typed in Devanāgarī, Bengali, Telugu, Kannada, Malayalam, Gujarati or Odia, or romanised with or without diacritics, reaches the same normal form and returns the same verses: a graphemic, rather than semantic, interlingua. The limit is diagnostic. Tamil does *not* fold and cannot: its script under-specifies voicing and aspiration, so *ka*, *kha*, *ga* and *gha* share one grapheme and the round trip loses information rather than being unimplemented. A graphemic interlingua reaches exactly as far as the writing systems are distinction-preserving.

Worked example. Bhagavad Gītā 2.47 resolves in the deployed graph to one Work realised by 26 independently addressable expressions — the Devanāgarī source, six translations, fifteen commentaries including Śaṅkara’s, Rāmānuja’s, Madhva’s and Abhinavagupta’s, a word-split and three recitations — rendered on demand into 173 script manifestations. One GET returns all of them with their provenance handles (CQ1).

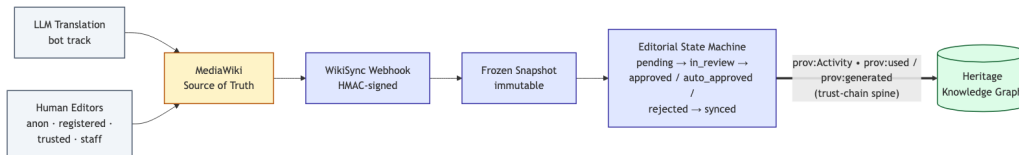


Figure 1: The trust-chain spine, left to right. Edits enter MediaWiki — from human editors at four trust levels and from the LLM translation bot — fire an HMAC-signed webhook, are frozen as immutable snapshots, and pass through the editorial state machine. Each transition emits a `prov:Activity` binding snapshot, actor and outcome into the graph, which realises each Verse Work as multiple Expressions (LRMoo · OntoLex-Lemon · PROV-O · SKOS). Figure 2 shows one exported chain.

3. The Governed Pipeline and the Trust Chain

Wiki as source of truth. All content is written on MediaWiki, in dedicated namespaces per lens. The wiki is open: anyone may comment on talk pages and propose edits. Authoritative content, however, is written by a small pool of trusted scholarly editors and moderators. This split — open at the contribution layer, gated at the promotion layer — is the social half of the trust model; the technical half is what follows.

The state machine. Every wiki save fires an HMAC-signed webhook into the service layer, which captures a frozen snapshot of the page content and opens a sync request. Requests move through a typed state machine (`pending` → `in_review` → `approved` → `synced`, with `rejected`, `superseded`, and `auto_approved` branches). Each transition is recorded as a `prov:Activity` with its actor, the snapshot it `prov:used`, and its outcome.

Trust tracks. Editors hold one of five trust levels. Anonymous and newly registered contributors require manual review; trusted editors get an editorial grace window; staff edits auto-approve. The fifth track, bot, exists for machine contributors: LLM-generated translation drafts enter here, are reviewable in batch, and are *never* auto-promoted. An AI draft becomes authoritative content only through an explicit human approval recorded in the chain. Given current evidence on hallucination in LLM output [10], we treat this as a hard invariant rather than a tunable policy.

Frozen snapshots. Approval binds to the snapshot that was reviewed, not to the live page. If the wiki moves on while a request is pending, the later edit produces its own request; the reviewer’s judgement is never silently re-applied to content they did not see.

The chain as a graph object. Every expression therefore answers, on inspection, what a sceptical reader would ask: who wrote this, at what trust level, what exactly the reviewer approved, when, and through which transition it entered the graph — via the REST API and audit log today, and as PROV-O activities in the OWL/RDF export, so a live SPARQL endpoint is a deployment step rather than a modelling one. Readers see the same chain as a provenance chip, and in full as a time-ordered timeline at `/demo/provenance/translation/553`.

4. The Chain as Triples

The claims above are worth only as much as the graph that backs them, so Figure 2 shows one, in full. It is the exported provenance of a single artifact — the Hindi translation of BG 2.47 — produced by walking the governance tables and serialising them through the vocabulary of Section 3. Nothing in it is hand-written: the activities are rows in the sync-request table, the agent is a wiki account with its trust level, and the two entities are the successive texts that were actually stored.

```
hssr:translation_553 a hss:Translation ;
  hss:ofVerse hssr:Verse_SBG_2_47 ;
  dcterms:language "hi" ;
  hss:lifecycleState "authoritative" ; hss:authorSlug "htrskd" .

hssr:editor_HSSTranslationBot a hss:WikiEditorProfile , prov:Agent ;
  foaf:accountName "HSSTranslationBot" ;
  hss:trustLevel hss:Trust_Bot .

hssr:syncrequest_6341 a hss:WikiSyncRequest , prov:Activity ;
  hss:wikiPage "Translation:SBG_2_47_htrskd" ;
  hss:requestState hss:State_Synced ; hss:trustTrack "ai" ;
  hss:editedBy hssr:editor_HSSTranslationBot ;
  hss:affectsLens hssr:translation_553 ;
  prov:startedAtTime "2026-04-25T12:01:18Z"^^xsd:dateTime .

hssr:revision_1 a hss:TranslationRevision , prov:Entity ;
  hss:revisionSource "ai" ; hss:lifecycleState "ai_drafted" ;
  hss:textSha256 "56ec2b4f4ee2fbc9...fbb0e93a" ;
  prov:wasGeneratedBy hssr:syncrequest_6341 .

hssr:reviewaction_1 a hss:ReviewAction , prov:Activity ;
  hss:action "promote" ; hss:verdict "approve" ;
  hss:reviewerName "Acharya" .

hssr:revision_2 a hss:TranslationRevision , prov:Entity ;
  hss:revisionSource "community_draft" ;
  hss:lifecycleState "authoritative" ;
  hss:textSha256 "befdd6e82b635d44...b914c1459" ;
  prov:wasGeneratedBy hssr:syncrequest_6342 ;
  prov:wasRevisionOf hssr:revision_1 .
```

Figure 2: Exported trust chain for the Hindi translation of BG 2.47, emitted by `export_provenance_rdf` against the production database (abridged; the full 80-line file accompanies the demo). The `prov:wasRevisionOf` edge answers “what changed”: an LLM draft on the `ai` track became authoritative only after a named reviewer’s promote, and both states remain.

5. Evaluation: Competency Questions on the Deployed System

We evaluate the schema by four competency questions, each chosen to probe work that a flat relational view of the corpus could not do directly: CQ1 exercises the multi-lens model (one Work, many independently addressable Expressions), CQ2 the trust chain itself, CQ3 the governance gate as a queryable fact, and CQ4 the cross-script normal form. CQ2 is the one the paper’s claim rests on, so we give it in full. Against the union of the T-Box and the exported chain of Figure 2:

```
SELECT ?rev ?source ?state ?track ?editor WHERE {
  ?rev a hss:TranslationRevision ;
    hss:revisionSource ?source ;
    prov:wasGeneratedBy ?act .
  ?act hss:requestState ?state ;
    hss:trustTrack ?track ;
    hss:editedBy ?editor . }
```

returns two rows — (`revision_1`, `ai`, `State_Synced`, `ai`, `editor_HSSTranslationBot`) and (`revision_2`, `community_draft`, `State_Synced`, `ai`, `editor_HSSTranslationBot`) — which is the history a sceptical reader wants: the current text descends from a machine draft, and the descent is visible rather than erased. Table 1 gives all four questions with the interface that answers each on live data; the demo video walks through CQ2 and CQ4 end to end.

Why involve a language model at all. A reasonable objection is that this architecture spends considerable machinery containing a risk it introduces. The answer is a throughput one. The scarce

Table 1

The four competency questions, the interface that answers each today, and the result on live data.

Question	Interface	Result
CQ1. Which expressions exist for BG 2.47, by which author, in which language?	GET /api/ SBG/2/47/	26 expressions — 1 source, 6 translations, 15 commentaries, 1 word-split, 3 recitations — and 173 script manifestations.
CQ2. Which chain produced the current text of this translation?	SPARQL over the exported chain (Section 4)	2 revisions; machine draft superseded by a promoted community draft, both retained.
CQ3. Which AI translations are still awaiting review?	GET /api/collab/ SBG/2.47/juxtapose/ ?lang=hi	The drafts held at ai_drafted on the bot track, none authoritative; 4,645 corpus-wide.
CQ4. Which verses match <i>karmāyevādhikāraṣṭe</i> , typed in any script?	GET /api/search/ ?q= . . .	The same 23 hits, BG 2.47 first, for all 8 folding scripts and both romanisations.

resource in this domain is not text but the attention of people qualified to judge it, and a reviewer handed a draft and asked *is this right* works at a different rate from one translating from nothing. The Vālmiki Rāmāyaṇa runs to tens of thousands of verses in eleven target languages; there is no staffing model under which that is produced by hand, so the alternative to a governed draft is not a careful human translation but no translation at all. The chain lets us make that trade without laundering its cost: a reader always sees which of the two they have.

What generalises beyond Sanskrit. Almost all of it. The governance pipeline, the trust tracks, the frozen snapshot, the PROV-O export and tradition-as-a-dimension are corpus-independent: they assume only a wiki, a review population stratified by trust, and a domain where sources legitimately disagree — which describes most textual heritage, and much of legal and scriptural scholarship besides. Two components are genuinely Sanskrit-specific: the cross-script normaliser, which exploits one language being written in many scripts that map to a common transliteration, and the sandhi-aware word splitting feeding the sub-lexical layer. A deployment elsewhere drops both and loses nothing else.

What the audit log measures, and what it does not. Every number here is counted on the production database. The governance tables hold 6,342 sync requests over 11 editor profiles spanning four trust levels; 56,141 translations, of which 4,645 stand at ai_drafted — LLM output that has entered the graph, is addressable and inspectable, and is not readable as authoritative content. The invariant that motivates the whole design is therefore checkable rather than asserted, and it holds: zero AI-sourced revisions carry the authoritative lifecycle state, so nothing has crossed the gate without a recorded human act. *Disposition rates* are a different claim and we do not make it: the review queue has processed one artifact end to end, so an approve/revise/reject ratio over $N=1$ would be a number with no measurement behind it. The log is populated by the core team, machine imports and a first limited-release cohort, not yet by sustained external expert review; onboarding those reviewers is the active next phase, and a scholar spot-evaluation of the drafts is in progress. We make no throughput or quality claim the deployed system cannot back today.

6. Conclusion and Roadmap

The interesting question about AI-assisted knowledge engineering is no longer whether models can draft useful content — they can — but what infrastructure lets a community accept that content on inspectable terms. HSS shows that a complete answer can be assembled today from standard parts: PROV-O over an editorial state machine, an open wiki under typed governance, and a corpus ontology in which tradition and language are dimensions rather than footnotes. Planned work includes a public

SPARQL endpoint over the OWL/RDF export (the modelling is in place; the deployment step is staged), a sense-disambiguated concept layer linking verses to a multilingual sense inventory (pre-release, so claimed as future work only), and agentic review assistance *inside* the bot track — AI helping to triage AI drafts, with the same invariant that nothing is promoted without a human in the chain.

Ethics and Environment

The corpus is public-domain classical text; translations and commentaries are used with attribution under their respective licences. The tradition-aware cascade is itself an ethical mechanism: no lineage is presented as canonical, and gaps are shown rather than papered over. Contribution is open at the wiki layer; promotion is gated by named, accountable reviewers. Environmental footprint is modest — LLM use is limited to draft generation feeding a human queue, with no recurring training workload, on a single host.

Declaration on Generative AI

In the system: LLMs generate translation drafts and seed sub-lexical analyses; all such output enters the knowledge graph on the bot trust track and is never promoted without human review. *In preparing this paper:* during the preparation of this work the authors used Anthropic Claude for outline structuring and copy-editing. After using this tool, the authors reviewed and edited the content, verified all factual claims against the deployed system and its database, and take full responsibility for the publication's content.

References

- [1] T. Lebo, S. Sahoo, D. McGuinness, et al., PROV-O: The PROV Ontology, W3C Recommendation, 2013. <https://www.w3.org/TR/prov-o/>.
- [2] P. Riva, M. Žumer, M. Doerr, et al., LRMoo: object-oriented definition and mapping from IFLA LRM, version 1.0, IFLA, 2024.
- [3] A. Miles, S. Bechhofer, SKOS Simple Knowledge Organization System Reference, W3C Recommendation, 2009. <https://www.w3.org/TR/skos-reference/>.
- [4] J. P. McCrae, J. Bosque-Gil, J. Gracia, et al., The OntoLex-Lemon model: development and applications, in: Proc. eLex 2017, pp. 587–597.
- [5] Göttingen Register of Electronic Texts in Indian Languages (GRETIL), Niedersächsische Staats- und Universitätsbibliothek Göttingen. <https://gretil.sub.uni-goettingen.de/>.
- [6] O. Hellwig, DCS — The Digital Corpus of Sanskrit. <https://www.sanskrit-linguistics.org/dcs/>.
- [7] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Commun. ACM 57 (10) (2014) 78–85.
- [8] M. Daquino, V. Pasqual, F. Tomasi, F. Vitali, Expressing without asserting in the arts, in: Proc. 18th Italian Research Conference on Digital Libraries (IRCDL 2022), CEUR-WS vol. 3160, 2022.
- [9] V. Pasqual, G. Manzano, E. Uzeir, F. Tomasi, F. Vitali, Evaluation of expressing without asserting approaches in RDF: the case of conjectures, in: Proc. 20th Italian Research Conference on Digital Libraries (IRCDL 2024), CEUR-WS vol. 3643, 2024, pp. 60–73.
- [10] Z. Ji, N. Lee, R. Frieske, et al., Survey of hallucination in natural language generation, ACM Comput. Surv. 55 (12) (2023) Art. 248.