Structured Context Retrieval

Structure-Preserving Retrieval for Trustworthy Enterprise AI

ARAMAI Research

July 2026

ABSTRACT

Retrieval-augmented generation was supposed to ground language models in an organization's own knowledge. In practice, the dominant implementation destroys that knowledge on the way in. Standard RAG pipelines fragment structured content into fixed-size chunks, discarding the hierarchies, types, and relationships that organizations spent decades encoding — and then ask a vector index to recover, by statistical similarity, the meaning the chunker just threw away. This paper describes **Structured Context Retrieval (SCR)**, a retrieval methodology built on a different premise: structure is meaning, and retrieval should preserve it rather than reconstruct it. SCR retrieves **components** — semantically meaningful units defined by their schemas, with type information and relationships intact — instead of **chunks**, arbitrary fragments defined by token counts. We define the component–chunk distinction, state the architectural commitments SCR entails, locate SCR within the current Graph RAG discourse by distinguishing schema-grounded graphs from LLM-extracted ones — a distinction that reverses the standard cost argument against graphs — and present the converging evidence: schema-aware retrieval reaching 92% accuracy against a 68% vector-only baseline, hallucination reductions on the order of 90%, and independent 2026 results that replicate the pattern. SCR is the retrieval discipline implied by the broader principle of consulting authority before generating — look up before you make up — which we develop in a companion paper.

Keywords: *structured context retrieval, retrieval-augmented generation, knowledge graphs, schema-grounded AI, Graph RAG, enterprise AI*

---

1. The Compression Nobody Audits

Every retrieval-augmented system makes a decision before the first query arrives: how to break source content into retrievable pieces. In the standard pipeline, that decision is made by a tokenizer. Documents are split into fixed-size chunks — 512 tokens, 1,000 characters, some window with overlap — because that is what embedding models and context budgets find convenient.

It is worth stating plainly what this step does to structured content. Take a fifty-page clinical protocol, a maintenance manual, a product catalog, a regulatory filing. Chunk it, and the section hierarchy is gone. Cross-references break. Tables dissolve into meaningless rows. A procedure's steps lose their sequence; a conditional statement loses the condition it depended on; a definition drifts away

from the term it defines. The fragments still read fine — each one is fluent in isolation — which is exactly why the damage goes unnoticed. The pipeline has performed a lossy semantic compression, and nothing downstream audits the loss.

The system is then asked to answer questions over these fragments by embedding similarity: find the pieces that sound most like the question, and hope the meaning survives. Sometimes it does. But for the questions enterprises actually care about — questions whose answers live in relationships, sequences, constraints, and state — similarity over fragments is structurally incapable of recovering what fragmentation destroyed. The failure is not in the model or the embedding quality. It happened at ingestion, and no amount of re-ranking repairs it.

2. Structure Is Meaning

The premise underneath SCR is simple to state. The hierarchical organization of content is not incidental formatting; it encodes semantics. A procedure's steps have sequence. A concept's sections have topical scope. A task's prerequisites have dependency. A table's rows have their columns. When organizations invest in structured content — technical documentation, healthcare records, product information, legal instruments — they are encoding domain expertise, business logic, and semantic relationships into that structure. Any retrieval system that destroys the structure destroys the meaning, however good its embeddings are.

The corollary is the distinction that does most of the work in this paper. A **component** is a semantically meaningful unit of content defined by its schema — possessing intrinsic boundaries, type information, and declared relationships, created by authors and systems with intent. A **chunk** is an arbitrary subdivision created by algorithmic tokenization, without regard for semantic boundaries or authorial intent. Its boundary means nothing; it is where the token counter happened to stop.

Characteristic	Component (SCR)	Chunk (Vector RAG)
Boundary origin	Semantic — authorial	Algorithmic — token count
Type information	Preserved	Lost
Relationships	Explicit, traversable	Implicit, inferential
Context recovery	Deterministic	Probabilistic
Provenance	Complete audit trail	Fragment reference

Table 1. Component versus chunk comparison.

Vector RAG retrieves chunks. SCR retrieves components.

3. What Structure-Preserving Retrieval Requires

SCR is a methodology, not a product feature, and it rests on a small number of architectural commitments, stated here at the level of principle. (The mechanisms that implement them are the subject of separate, protected work.)

— **Let the schema organize the knowledge.** Well-designed content schemas already encode a domain's ontological commitments — its types, its containment relationships, its constraints. SCR treats the schema as the organizing principle of the retrievable knowledge, rather than constructing an organization from scratch or inferring one from raw text. The consequence is schema agnosticism: the methodology applies to any content structure whose units are typed, addressable, explicitly related, and capable of carrying metadata — DITA and DocBook in documentation, FHIR in healthcare, Schema.org in commerce, XBRL in finance, Akoma Ntoso in law. None of these standards is a requirement; each is an exemplar of the same universal property.

— **Discover by similarity; retrieve by structure.** Vector search and structural traversal are good at different things, and SCR uses each for what it is good at. Semantic similarity is excellent at finding candidates — surfacing the regions of a knowledge base relevant to a question, including phrasings no keyword would match. It is unreliable at delivering context — the retrieved fragment arrives stripped of its neighbors, its type, and its constraints. So in SCR, similarity identifies where to look, and the structured knowledge graph delivers what was actually there: the full component, with its relationships and metadata intact.

— **Make provenance deterministic.** Every generated answer should be traceable, through explicit graph relationships, to the components that grounded it. This is more than a compliance feature; it is an epistemological commitment. Knowledge has sources; claims have evidence. A system that can show which authoritative components produced an answer can be audited, corrected, and trusted. A system that can only gesture at “the passages that were nearby in embedding space” cannot.

— **Respect content state.** Structured content carries governance metadata at every level — publication status, version, expiration, entitlement, locale. Because SCR's units are components rather than fragments, that state travels with them, and retrieval can be filtered by it. This closes a family of failures endemic to chunk-based systems: serving expired content, mixing draft with published material, violating entitlements, answering in the wrong language. In regulated environments these are not edge cases; they are the requirements.

4. Three Sources of Graph Truth

The industry conversation has, over the past two years, converged on graphs — Graph RAG in its many variants is now the acknowledged answer to vector RAG's relational blindness. But the conversation routinely conflates approaches that differ in the property that matters most: where the graph's truth comes from.

LLM-extracted graphs. The best-known Graph RAG systems build their graph by running a language model over the text corpus — extracting entities and relations, clustering them into

communities, summarizing the clusters. This works, and it is the right tool when the corpus is genuinely unstructured. But it has two structural costs. First, price: the extraction pass is enormously expensive — indexing costs commonly cited at one to two orders of magnitude above baseline RAG, which is precisely the overhead subsequent variants exist to reduce [1]. Second, and more fundamental, epistemic status: an extracted graph is a statistical estimate of the corpus's structure. Its nodes and edges inherit the extractor's errors and ambiguities, and no reasoner can make the result authoritative after the fact.

Manually engineered ontologies. At the other pole sits hand-built ontology work: authoritative, and famously slow and expensive to construct and maintain — the classical knowledge-engineering bottleneck.

Schema-grounded graphs. The third source — the one the current discourse keeps reaching for without naming — is the structure organizations already possess. Content schemas, data models, and canonical definitions are authored structure: deliberate, governed, already fought over in meetings. Deriving the knowledge graph from these schemas yields a graph that is authoritative by construction, at a fraction of the cost of either alternative, because the expensive intellectual work — deciding what the types and relationships are — was completed years ago. SCR is retrieval over this third kind of graph.

This reframing reverses the standard cost argument. The 10–100× indexing overhead attributed to “graphs” belongs specifically to text-first extraction — to reconstructing, by inference, structure that was never captured. Where structure already exists, the graph is not an expense to be justified; it is an asset being recovered. The premature compression happens once, at ingestion, and every downstream system pays for it forever. SCR simply declines to compress.

5. The Evidence

The case for structure-preserving retrieval is now empirical, and it converges from independent directions. Across our production deployments, error falls sharply at the moment the system retrieves components instead of chunks — because the answer-bearing relationships arrive intact instead of being re-guessed.

Metric	Result
Schema-aware retrieval accuracy	92%, against a 68% vector-only baseline on identical workloads [2]
Hallucination reduction	~90% on tested production workloads [2]
Independent replication (KDD 2026)	Typed knowledge-graph grounding on an industrial-operations agent: 65% → 99% [3]

Table 2. Summary of empirical results.

The replications extend beyond us. Anthropic's published account of its internal analytics agents reports 21% accuracy without a curated structural layer and 95% with it — and a documented drift back to 65% when the layer fell one month behind the data [4]. A dedicated academic treatment concludes that governed semantic structure is the dominant variable in hallucination mitigation across frontier models [5]. Even model architecture is converging: DeepSeek's Engram work separates pattern lookup from dynamic reasoning inside the network itself — the same asymmetry SCR exploits at the system level [6]. And on multi-hop questions, where answers traverse six or more relationship links, the gap becomes categorical: a structured graph performs the traversal deterministically; similarity over fragments performs it by luck [7].

6. Scope, Honestly Stated

A methodology earns trust by declaring its boundaries. SCR presupposes that structure exists. For a corpus of genuinely unstructured prose with no schema and no implicit model worth recovering, LLM-assisted extraction is the right tool, and vector retrieval alone may be entirely adequate for similarity-shaped questions over it. SCR's claim is narrower and, for enterprises, more consequential: **where structured content exists, destroying its structure at ingestion is an unforced error** — and most high-stakes enterprise content is structured, because regulation, safety, and interoperability demanded it long before AI did.

SCR also inherits a governance obligation. A schema-grounded graph is authoritative only while its schemas are current; structure, like the semantic definitions above it, must be maintained as a product with an owner, not extracted once and abandoned. Retrieval discipline without governance discipline degrades silently.

7. Retrieval as the Second Half of a Principle

SCR does not stand alone. It is the retrieval half of a broader order of operations we have argued for elsewhere: look up before you make up — consult the authoritative structure that governs a question before the model generates, and let what is found constrain the answer [8]. Looking up is only as good as what the lookup returns. A system that consults authority but receives fragments has honored the principle in name and lost it in transit. SCR is what makes the lookup worth performing: the structure consulted arrives as structure, with its types, relationships, state, and provenance intact.

The conclusion of the argument is short. Organizations do not need to manufacture meaning for their AI systems; they need to stop destroying the meaning they already have. Retrieve components, not chunks. Let the schema organize the knowledge. Discover by similarity, retrieve by structure, and trace every answer to its source. Structure is meaning — and preserved structure is the difference between a system that sounds grounded and one that is.

References

[1] Microsoft Research. *GraphRAG* (2024) and *LazyGraphRAG* (2024–2025): LLM-extracted graph indexing and the cost-reduction variants it motivated.

[2] ARAMAI production measurements, schema-aware retrieval vs. vector-only baseline, 2025–2026 deployments.

[3] “AssetOpsBench: Typed Knowledge-Graph Grounding for Industrial-Asset LLM Agents.” arXiv:2605.26874, KDD 2026.

[4] Anthropic. “How Anthropic Enables Self-Service Data Analytics with Claude.” claude.com, June 2026.

[5] Rumiantsau, M., et al. “Semantic Layers for Reliable LLM-Powered Data Analytics.” arXiv:2604.25149, 2026.

[6] DeepSeek. *Engram: Conditional Memory as an Architectural Primitive.* January 2026.

[7] Promethium. Knowledge-graph substitution evaluation, 2026.

[8] ARAMAI. “Look Up Before You Make Up: Schema-Grounded Generation for Trustworthy Enterprise AI.” aramai.net/resources, 2026.