10.48047/jocaaa.2023.31.04.68

GenAI-Powered Knowledge Graphs for Enterprise Intelligence and Secure Data Interoperability

Sree Kiran Kanchanapally
Business Intelligence Developer
kanchanapallysreekiran@gmail.com

Abstract

Enterprises increasingly rely on generative AI, yet fragmented schemas, siloed stores, and inconsistent governance limit factuality, traceability, and secure cross-domain exchange. This paper proposes a GenAI-powered Enterprise Knowledge Graph (EKG) stack that tightly couples large language models (LLMs) with standards-based semantics (RDF/OWL/SHACL) and policy-aware retrieval-augmented generation (RAG) under zero-trust principles. first synthesize the state of LLM–KG integration—including subgraph-level retrieval, entity/path grounding, and explanation-aware prompting—and identify design gaps in provenance, access mediation, and multi-tenant governance. then specify a reference architecture that aligns W3C RDF 1.2/SPARQL 1.2 for interoperable graph access, SHACL for constraint validation, and a security plane that combines OAuth 2.1/OIDC with Verifiable Credentials (VC) 2.0 to enable claim-based, least-privilege authorization across organizational boundaries. The methodology details ontology engineering, R2RML/CSVW mappings, SHACL-driven data quality, named-graph segmentation for policy scoping, and dual indexing (symbolic + lexical) for KG-RAG with citation-backed answers. This work propose evaluation protocols spanning (i) answer quality and faithfulness with provenance checks, (ii) graph quality via constraint-violation and competency-coverage metrics, (iii) privacy leakage through red-teaming and canary entities, and (iv) zero-trust posture via policy-decision accuracy and audit completeness. Two condensed case illustrations—supplier-risk exchange and manufacturing FMEA analytics—demonstrate improved factuality, explainability, and controlled data minimization compared to naïve RAG. conclude with deployment guidance for regulated settings, highlighting limitations (ontology stewardship, latency-cost trade-offs, policy sprawl) and a roadmap toward provenance-aware evaluation and confidential computing for sensitive federations. The results indicate that converging GenAI with standards-based EKGs yields verifiable, secure, and updatable enterprise intelligence ready for production use in compliance-intensive contexts.

Keywords: Enterprise Knowledge Graph; Retrieval-Augmented Generation; Verifiable Credentials; Zero-Trust Architecture; Data Interoperability.

10.48047/jocaaa.2023.31.04.68

1. Introduction

Enterprises today generate and consume data across an expanding constellation of systems—transactional databases, data lakes and lakehouses, document repositories, IoT telemetry, SaaS applications, partner portals, and external data marketplaces. The resulting heterogeneity fragments business context and obscures the relationships among customers, products, assets, processes, controls, and risks. Conventional integration approaches (e.g., schema-on-read catalogs or ETL pipelines) often normalize formats but fail to harmonize meaning, leaving analysts to reconcile semantics ad hoc. Knowledge graphs (KGs) address this gap by representing entities and relationships as first-class citizens, enabling a unified, semantically consistent view that supports entity resolution, lineage tracking, and machine reasoning across silos. In large organizations, enterprise knowledge graphs (EKGs) become a shared, governed substrate that aligns operational systems with analytics, governance, and decision automation.

In parallel, the emergence of generative AI—particularly large language models (LLMs)—has transformed how users query, summarize, and synthesize information. LLMs excel at natural-language understanding, instruction following, and content generation, drastically reducing the friction between business users and complex data estates. Yet, when applied directly to enterprise scenarios, vanilla LLMs face four persistent limitations: (i) freshness, because model parameters lag behind evolving data and policies; (ii) factuality, as generative outputs can misstate or overgeneralize; (iii) provenance, since answers seldom carry verifiable citations to authoritative sources; and (iv) policy enforcement, because authorization and purpose limitations must be applied at retrieval and composition time, not merely at training time. These constraints are particularly acute in regulated domains—financial services, healthcare, critical infrastructure—where errors and policy violations carry high operational and compliance risk.

Integrating LLMs with knowledge graphs offers a principled path to overcome these constraints. In KG-augmented generation (often implemented as retrieval-augmented generation, or RAG), the model is grounded with up-to-date, curated graph context—entities, relationships, and constraints—prior to generation, improving factuality and enabling explanations anchored in graph paths. In LLM-augmented KGs, models assist human stewards with ontology extension, entity linking, and reference reconciliation, accelerating graph construction while retaining human-in-the-loop governance. Hybrid, joint frameworks coordinate both directions: the KG constrains and explains model outputs, and the model

10.48047/jocaaa.2023.31.04.68

bootstraps and maintains the KG at scale. Within these patterns, subgraph-level retrieval and explanation-aware prompting have emerged as effective design choices: rather than feeding long, unstructured documents to the model, systems retrieve compact, policy-cleared subgraphs and document snippets that precisely support the user task, then surface the retrieval traces as auditable explanations.

Despite these advances, enterprise adoption hinges on secure, interoperable operation across organizational boundaries. Data sharing increasingly spans subsidiaries, suppliers, distributors, service providers, and regulators—each with distinct policies, contracts, and jurisdictions. To support such collaboration, the stack must align with web-native semantics and identity standards. RDF and SPARQL provide a vendor-neutral graph data model and query language for interoperable access; OWL and SHACL capture semantics and constraints to ensure quality and consistency; modern authorization profiles such as OAuth and OpenID Connect provide robust authentication and delegated authorization; and verifiable, cryptographically signed credentials allow portable trust assertions about organizations, people, devices, and datasets. Anchoring the stack in Zero-Trust Architecture (ZTA) further requires explicit verification on every request, least-privilege access, continuous monitoring, and segmentation at the data and service layers. Together, these standards and practices enable machine-verifiable, policy-aware exchange of knowledge across domains.

From a systems perspective, enterprises need more than an ad hoc coupling between a vector index and a language model. They require a reference architecture that cleanly separates ingestion and mapping, semantic storage, security and policy, GenAI reasoning, and cross-domain exchange. Ingestion must support diverse sources (RDBMS, files, event streams, APIs) with reproducible mappings; the semantic core must implement graph storage, ontology management, and constraint validation; the security plane must enforce dynamic, attribute-based policies and capture provenance; the reasoning layer must combine symbolic retrieval (SPARQL, path queries) with lexical retrieval (BM25, embeddings) and generation, preserving citations; and the interoperability layer must support federation and credential-gated sharing with auditable logs. Observability and governance span all layers, with metrics for data quality, access decisions, model faithfulness, cost, and latency.

Equally important is a rigorous evaluation methodology. Beyond traditional IR or QA accuracy, enterprise GenAI must be assessed on faithfulness and provenance (does the answer follow from retrieved facts, and are sources cited?), graph quality (constraint violations,

10.48047/jocaaa.2023.31.04.68

duplication, and coverage against competency questions), security posture (accuracy of policy decisions, detection of misconfigurations), and privacy leakage (resistance to prompt injection, data exfiltration, and membership inference). Operational concerns—latency, throughput, and cost efficiency—must be measured under realistic workloads, including multi-tenant traffic and federated queries that cross trust boundaries.

This paper contributes a GenAI-powered EKG stack that operationalizes these requirements for enterprise intelligence and secure data interoperability. Also synthesize recent patterns in LLM–KG integration; define a standards-aligned architecture that embeds zero-trust controls; propose design patterns for lineage, access control, and multi-domain federation; and outline reproducible evaluation protocols spanning quality, robustness, privacy, and auditability. illustrate the approach with representative scenarios—supplier risk exchange and manufacturing analytics—where policy-aware KG-RAG improves factuality, explainability, and data minimization relative to naïve prompt-augmented systems. Finally, discuss limitations (ontology stewardship, prompt brittleness, policy sprawl, and latency–cost trade-offs) and chart a roadmap toward confidential computing for sensitive joins, provenance-aware scoring, and automated policy testing at scale.

By converging generative models with governed, standards-based knowledge graphs—and embedding zero-trust and verifiable identity throughout the pipeline—enterprises can transform fragmented data into reliable, explainable, and compliant intelligence that scales across teams and partners, while preserving security and privacy by design.

2. Background and Related Work

Knowledge graphs and standards

Knowledge graphs in enterprises have evolved from ad hoc link repositories to governed semantic substrates formalized by open standards [1]. RDF 1.2 defines the abstract data model for interoperable graph storage and exchange, clarifying graph naming, literals, and dataset constructs while maintaining backward compatibility [2]. RDF Schema adds a lightweight vocabulary for classes and properties, enabling reusable domain taxonomies that tools can validate and reason over [3]. SPARQL 1.2 advances federation, property paths, and entailment alignment so distributed graphs can be queried as a virtual whole without forcing a single physical consolidation [4]. For quality assurance, SHACL provides executable “shapes” that turn data contracts into machine-verifiable constraints [5].

10.48047/jocaaa.2023.31.04.68

LLM–KG integration

Recent work converges on three complementary patterns for combining generative models with knowledge graphs [6]. First, KG-augmented generation grounds prompts with entities, relations, and constraints retrieved from the graph (often via retrieval-augmented generation), improving factuality and enabling verifiable, path-based explanations [7]. Newer methods emphasize subgraph-level retrieval to provide compact, topology-aware context that aligns with the user's question [8]. Second, LLM-augmented KGs use models to accelerate schema extension, entity linking, and reference reconciliation with human-in-the-loop review, scaling graph construction while retaining governance [9]. Third, joint frameworks iterate in both directions—graphs constrain and explain model outputs, while models propose graph updates—yielding systems that are more current and controllable than either component alone, with empirical findings showing reduced hallucinations and increased answer faithfulness via graph-aware retrieval [10].

Security and interoperability

Secure, cross-domain operation is increasingly framed by Zero-Trust Architecture, which mandates explicit verification on every request, least-privilege access, continuous monitoring, and segmentation at data and service layers [11]. Within identity and authorization, OAuth 2.1 consolidates modern best practices for delegated access [12], while OpenID Connect supplies an identity layer and claims model to bind users and contexts to graph queries [13]. For portable, cryptographically verifiable assertions across organizations, Verifiable Credentials 2.0 standardizes signed, privacy-preserving claims such as roles, certifications, and attributes [14], and OpenID for Verifiable Presentations specifies how digital wallets present those credentials to relying parties [15]. In combination, these standards enable policy-aware interoperability: SPARQL federates queries across domains; zero trust governs the decision points; OAuth/OIDC convey and verify access; and VC-based claims allow fine-grained, minimal disclosure aligned with purpose limitation, all with auditable end-to-end provenance.

3. Problem Statement

Modern enterprises operate across heterogeneous, fast-changing data estates where critical information is scattered among relational databases, data lakes, documents, SaaS systems, IoT streams, and partner platforms. This fragmentation prevents generative AI from relying on a single, trusted source of truth, leading to stale answers, hallucinations, and inconsistent decisions. The core problem is to establish a unified intelligence fabric that grounds GenAI in

10.48047/jocaaa.2023.31.04.68

verified, governed knowledge while preserving the semantics, lineage, and contractual obligations of each source. Such a fabric must expose business entities and relationships in a machine-interpretable form, continuously synchronize with upstream systems, and make every AI response traceable to authoritative evidence.

At the same time, enterprises must enforce fine-grained, context-aware access control at the exact moment of retrieval and generation. Policies vary by jurisdiction, data class, user role, purpose, and consent; they also evolve with regulations and internal risk posture. Static, perimeter-based controls are insufficient when models can compose information across domains. The challenge is to evaluate dynamic authorization including attribute-, purpose-, and context-based constraints on every query and generation step, minimize data exposure, and produce verifiable logs that withstand audits. This must hold across multi-tenant deployments and federated queries that span subsidiaries, vendors, and regulators without centralizing all data or diluting local autonomy.

Enterprises also face interoperability and portability requirements. Business value increasingly depends on sharing specific, minimal slices of knowledge with partners supplier risk attestations, product-safety facts, service entitlements under explicit provenance and consent. The problem is to enable cross-organization exchange where claims about identities, certifications, and data quality are cryptographically verifiable; where schemas differ yet can be aligned; and where usage can be purpose-limited and revoked. This calls for standards-based semantics and identity so that models can reason over heterogeneous graphs while honoring contractual data-sharing obligations and regional data-residency constraints.

Finally, the solution must be explainable and auditable by design. Risk, compliance, and domain experts require answers that are not only correct but provably derived from governed sources, with clear retrieval traces, policy decisions, and model prompts preserved for review. Non-functional constraints latency, cost, scalability, robustness to prompt injection and data leakage are first-class requirements. In sum, the problem is to design and operationalize a GenAI-powered knowledge layer that unifies semantics, enforces dynamic policy, enables secure interoperability with provenance and consent, and yields transparent, regulator-ready explanations without sacrificing performance or maintainability.

4. Methodology

Overall approach.

10.48047/jocaaa.2023.31.04.68

The methodology follows a lifecycle that begins with domain scoping and competency questions, proceeds through ontology engineering and data onboarding, and culminates in policy-aware KG-RAG (knowledge-graph–grounded retrieval-augmented generation) with continuous evaluation. The objective is to operationalize a governed knowledge layer that reliably grounds generative answers in verifiable enterprise facts while enforcing dynamic authorization and preserving end-to-end provenance.

Domain scoping and competency questions.

And begin by convening domain owners from risk, operations, product, compliance, and data governance to define a bounded but high-value scope (e.g., supplier risk, product lifecycle, incident response). For each scope, and elicit competency questions—precise, testable queries the system must answer, such as "Which suppliers with critical components lack a current certification for region X?" or "What failure modes are linked to this asset across plants?" These questions drive ontology design, mapping priorities, access policies, and RAG prompt patterns. Each competency question is paired with acceptance criteria (e.g., coverage thresholds, latency budgets) and tagged with sensitivity classes to shape policy and evaluation.

Ontology engineering and governance.

And construct a modular enterprise ontology that separates core, domain, and extension modules. The core captures cross-cutting entities (Organization, Person, Asset, Product, Risk, Control, Evidence) and relations (owns, supplies, mitigates, cites, derivedFrom). Domain modules codify sector-specific semantics (e.g., FMEA, CAPA, regulatory attestations), while extension modules house local customizations to avoid polluting shared semantics. Ontology stewardship is formalized via a schema council and a change-control workflow: proposals are submitted as pull requests with rationale, examples, and impact analysis; automated checks verify naming, reuse of existing terms, and backward compatibility; and versioning is maintained with release notes that downstream teams can track. Competency questions are mapped to ontology fragments to ensure every question has explicit semantic coverage.

Source system inventory and data contracts.

Also catalogue authoritative sources across RDBMS tables, data lakes, document repositories, SaaS APIs, event streams, and partner feeds. For each source, also define a data contract specifying schema, refresh cadence, trust level, lineage attributes, and access class (public, internal, confidential, restricted). Sources are prioritized by their contribution to competency

10.48047/jocaaa.2023.31.04.68

questions and their sensitivity. Where possible, upstream owners expose stable views (for RDBMS) or curated zones (for data lakes) to minimize downstream breakage. Contracts also enumerate redaction rules (e.g., PII masking) and purpose tags required for access decisions.

Mapping and ingestion pipelines.

Relational sources are mapped to the ontology using repeatable mapping specifications (e.g., R2RML-style rules) that transform rows into entities and relationships while attaching provenance (source table, extract timestamp, transformation hash). File-based inputs (CSV, JSON, XML, PDF-derived text) are onboarded via declarative schemas and parsing recipes that normalize datatypes, units, and codes. API and event sources are ingested through streaming connectors that buffer and validate messages before materializing them as graph updates. All mappings are expressed as code-reviewed artifacts stored in version control, accompanied by synthetic test datasets and golden outputs to support continuous integration.

Data quality and constraint validation.

And define machine-readable quality rules as shapes and constraints that check cardinalities, value ranges, closed sets, datatype conformance, and referential integrity. During ingestion, batches are validated; violations generate metrics and are routed to a triage queue with sample rows and suspected root causes. For chronic violations, either fix upstream issues (preferred) or codify deterministic remediation steps (e.g., unit conversion, code system mapping) with explicit provenance. A quality dashboard reports violation rates, entity duplication, and coverage of key attributes, enabling stewards to track improvement over time.

Entity resolution and reference reconciliation.

To unify records across silos, and combine deterministic keys (e.g., DUNS, GTIN, serial numbers) with probabilistic and LLM-assisted matching. The pipeline computes blocking keys and similarity features (token, phonetic, embedding-based), proposes candidate merges, and routes low-confidence cases to human review. Every merge decision is stored as a first-class artifact with justification, supporting reversibility and audit. Where identifiers are ambiguous across jurisdictions,also maintain equivalence sets and scoped identity graphs, avoiding unsafe global merges.

Graph storage, partitioning, and lineage.

10.48047/jocaaa.2023.31.04.68

The enterprise knowledge graph is materialized in a quad store with named-graph partitioning by domain, region, and sensitivity. Operational and historical snapshots are separated to balance query performance and full lineage retention. All ingested statements carry provenance links to source datasets, transformation steps, and validation outcomes. This design enables subgraph extraction that includes both factual edges and the minimal provenance required for audit and explanation.

Security and policy enforcement.

Access control is enforced at query time using a policy engine that evaluates attribute- and context-based rules. Subjects (users, services) carry claims about roles, business units, regions, and purposes; resources (graphs, entities, predicates) carry labels for sensitivity, residency, and contractual terms; environment attributes include request time, network context, and assurance level. Policies are expressed as testable rules that permit or deny access to triples, subgraphs, and even attribute values (e.g., return a redacted label if full access is not permitted). Tokens presented by callers are verified, scopes are mapped to allowable operations, and purpose tags are checked against consent and contractual constraints. All decisions produce structured audit events linked to the returned subgraph.

KG-RAG retrieval design.

Retrieval is dual-channeled: a symbolic path retriever uses graph queries to collect minimal, topology-consistent subgraphs relevant to the competency question; in parallel, a lexical retriever indexes unstructured evidence (documents, notes, reports) associated with those entities. A **subgraph packer** prunes and formats the result into a compact context bundle containing canonicalized triples, key attributes, and citations to authoritative evidence. The packer enforces policy by stripping prohibited predicates and replacing sensitive literals with redacted tokens when necessary. This structured bundle becomes the grounding context for the generator.

Prompting, generation, and explanation capture.

Prompts are templatized per task (QA, summarization, classification, rationale generation) and include the subgraph bundle, enumerated facts, and explicit instructions to only use provided context. The generator is required to return: a natural-language answer, a list of cited entities and evidence URIs, and a brief rationale that references graph relations (e.g., “Supplier S is non-compliant because lacks relation hasCertification→ISO-27001 validThrough ≥ today”).

10.48047/jocaaa.2023.31.04.68

The system stores the prompt, retrieved subgraph identifiers, model parameters, and the produced answer in an immutable log, enabling deterministic replay for audit.

Privacy preservation and data minimization.

Before any retrieval or generation, the request is evaluated for purpose limitation; only attributes strictly necessary to answer the question are fetched. For analytics over sensitive cohorts, differential privacy can be applied at aggregation time to bound disclosure risk. In cross-organization exchanges, minimal-disclosure credentials are requested (e.g., "over-18" rather than full birthdate; "valid ISO-certified supplier" rather than full certificate body), and proofs are verified prior to query execution. Sensitive joins across domains can be performed inside isolated compute enclaves where feasible, with only aggregate results leaving the enclave.

MLOps, dataops, and policy-as-code.

All artifacts—ontologies, mappings, constraints, prompts, retrieval templates, and policies—are versioned and promoted across environments via continuous integration and delivery. Unit tests validate mappings and constraints; regression tests replay a library of competency questions and compare answers, citations, and decision logs against golden baselines. Canary deployments meter a small fraction of traffic to new retrieval or prompt versions and monitor faithfulness, latency, and cost before full rollout. Policy changes ship with test suites that encode expected allow/deny outcomes for representative requests.

Observability and SRE.

End-to-end telemetry captures ingestion throughput, constraint violations, entity merge rates, cache hit ratios, retrieval time, model time, and total answer latency. A faithfulness monitor samples answers and automatically checks whether cited facts entail the claim set. Security telemetry records policy evaluations, denials, and anomaly signals (e.g., unusual query patterns). Dashboards surface service-level objectives: P95 latency, cost per successful answer, retrieval success rate, and violation budgets for hallucination and policy errors. Alerts trigger runbooks that include fast rollback procedures for prompts, retrievers, and policies.

Evaluation protocol.

Evaluation spans four axes aligned to enterprise requirements. Quality is measured with exact-match or F1 on structured competency questions, human judgments on factual support, and

10.48047/jocaaa.2023.31.04.68

contradiction detection against the retrieved fact set. Graph quality is tracked via constraint-violation rate, duplicate-entity rate, and competency coverage. Security posture is assessed by policy-decision accuracy on a labeled corpus of allow/deny cases, denial-of-service resilience, and access-scope correctness under token variations. Privacy leakage is probed through red-team prompts, canary records, and membership-inference stress tests; acceptable thresholds are defined with risk and compliance teams. All metrics are reported per domain and per sensitivity tier to prevent averages from hiding high-risk clusters.

Case study execution.

For each target domain (e.g., supplier risk; manufacturing analytics), and instantiate the pipeline end-to-end. also select a representative set of sources and define domain-specific shapes, mappings, and policies. then implement five to ten high-value competency questions, seed golden answers with ground truth from domain stewards, and run A/B comparisons between (a) KG-RAG with subgraph packing and (b) document-only RAG baselines. And measure accuracy, faithfulness, latency, and cost, and perform ablations that remove policy checks or provenance to quantify their effect on performance and risk.

Deployment and scalability considerations.

The production topology separates hot path services (policy evaluation, retrieval, generation) from batch ingestion and heavy analytics. Caching is applied to stable subgraphs and authorization results with short TTLs; query planners use statistics to push down filters and avoid over-expansion. Horizontal scaling is achieved by sharding named graphs by domain and region, while the prompt/generation layer autoscale based on queue depth and latency targets. Cost controls include adaptive context sizing, answer summarization tiers, and model-selection heuristics that route simple queries to smaller models.

Risk management and compliance alignment.

Before go-live, a controls matrix maps pipeline steps to organizational policies and regulatory requirements (records management, access logging, data residency, incident response). Tabletop exercises simulate policy misconfiguration, credential misuse, and prompt-injection scenarios; findings drive additional guards, such as input validation, content filters, deny-lists for high-risk predicates, and approval gates for graph updates proposed by models. A periodic review cycle recalibrates policies, prompts, and shapes as regulations and business requirements evolve.

10.48047/jocaaa.2023.31.04.68

Maintenance and continuous improvement.

Post-deployment, a weekly governance cadence reviews drift in ontology and data quality, evaluates new competency questions from business stakeholders, and triages observed hallucinations or policy denials. Feedback loops close the gap between user behavior and system design: frequently denied requests inform policy refinements or new data contracts; recurring hallucination patterns inform shape or ontology extensions; latency regressions trigger retrieval tuning or index redesign. This continuous improvement cycle ensures the KG-RAG stack remains accurate, explainable, secure, and aligned with evolving enterprise needs.

4. GenAI-Powered EKG Stack

Layers and components

Ingestion & Mapping.

The stack begins with durable connectors that normalize heterogeneous sources—relational databases, data lakes, SaaS APIs, document repositories, and event streams—into logical staging models. Relational views are mapped to the enterprise ontology using repeatable, version-controlled rules (R2RML-style), while file and semi-structured inputs use schema descriptors akin to CSVW to standardize datatypes, units, and code systems. Every transformation attaches provenance (source, extraction time, transform hash) and is validated against machine-readable constraints (SHACL shapes) during load. Failed records are quarantined with diagnosable error messages so upstream stewards can correct issues without blocking the entire pipeline.

Semantic Core.

Curated data are materialized in an RDF 1.2-compliant store that supports both triples and quads, enabling named-graph partitioning by domain, region, sensitivity, or tenant. Core semantics are captured in RDFS/OWL ontologies with modular layering (core, domain, extension) to promote reuse and safe evolution. Query access is exposed via SPARQL 1.2 endpoints that support federation and service clauses, allowing the system to resolve questions across distributed graphs without physically consolidating every dataset. This layer is the "source of semantic truth," supplying compact, topology-consistent subgraphs to the reasoning layer and guaranteeing line-of-sight to origin and lineage.

Security & Policy Plane.

10.48047/jocaaa.2023.31.04.68

Authentication and authorization are enforced at request time using modern, token-based flows. Clients authenticate through OpenID Connect and receive OAuth 2.1 tokens with strong client protections (e.g., PKCE, MTLS or DPoP for token binding). Fine-grained ABAC/RBAC policies bind subjects (users, services) to resources (graphs, entities, predicates) under specific purposes, jurisdictions, and sensitivities; these policies are evaluated before any triple is returned. In line with zero-trust principles, every call is verified explicitly, policies are enforced per request, telemetry is captured continuously, and data and services are micro-segmented to reduce blast radius. To enable portable trust and auditable assertions, verifiable credentials convey issuer-signed attributes such as supplier certifications or device attestations, while PROV-O lineage links every returned fact to its sources, transforms, and validations.

GenAI Reasoning.

The reasoning layer operationalizes KG-RAG. Instead of feeding long, unstructured text to a model, a symbolic retriever assembles minimal subgraphs—entities, relationships, and key attributes—directly relevant to the question; a parallel lexical retriever brings in linked evidence from documents. A subgraph packer merges these results into a compact, policy-cleared context bundle with canonicalized facts and citations. Prompts instruct the model to reason only over this bundle, generate answers, and return the supporting citations. In the opposite direction, LLM-for-KG assists stewards by proposing schema merges, entity linking, and reference reconciliation, but changes are staged for human approval with automated checks to prevent semantic drift. The system records retrieval paths, applied policies, and short natural-language rationales (explanation-aware prompting), so every answer is auditable.

Interoperability & Exchange.

Cross-enterprise collaboration relies on standards-aligned federation and credential-gated exchange. Partners can query only the minimal triples necessary to fulfill a declared purpose, either through SPARQL federation across organizational boundaries or through payloads guarded by verifiable credentials. Consent and purpose binding are evaluated at runtime; results are encrypted in transit and at rest; and for high-sensitivity joins, optional confidential-computing enclaves confine processing, releasing only policy-approved aggregates or redacted values.

Observability & Governance.

10.48047/jocaaa.2023.31.04.68

The stack emits end-to-end telemetry. Data quality metrics track coverage, constraint violations, duplication, and drift. RAG telemetry monitors retrieval hit rates, subgraph sizes, generation latency, and hallucination flags based on faithfulness checks. Access audits capture who requested what, under which policy, with which credentials, and why it was allowed or denied. Governance dashboards summarize zero-trust posture, policy effectiveness, and change history for ontologies, mappings, prompts, and policies. These signals feed SLOs for latency, accuracy, cost, and security, with alerting and rollback playbooks.

Control flows (illustrative)

Secure Question Answering.

A user authenticates through OpenID Connect and obtains an OAuth 2.1 token scoped to specific operations and purposes. The request hits the policy engine, which evaluates subject attributes (role, business unit, region), resource labels (sensitivity, residency, contractual terms), and environment context (time, assurance level). If permitted, the symbolic retriever issues SPARQL queries to extract a minimal, policy-compliant subgraph; the lexical retriever optionally adds linked evidence snippets. The subgraph packer assembles a compact, canonical context and strips disallowed predicates or values. The LLM receives this bundle and generates an answer that must list the cited entities and evidence and provide a brief rationale referencing graph relations. The system writes an immutable log containing the request, policy decision, retrieved subgraph identifiers, prompt, model parameters, answer, citations, and confidence signals—supporting replay, audit, and red-team analysis.

Cross-Organization Data Share.

A partner begins by presenting a verifiable credential that attests to specific attributes (for example, “certified supplier for product line X through date Y”). The verifier checks signature integrity, issuer trust, and revocation status, then issues a constrained access token that encodes permitted purposes and scopes. The partner executes a federated SPARQL query or calls a credential-gated API; the policy engine enforces purpose limitation and data minimization, authorizing only the smallest subgraph required to answer the request. If the query spans sensitive domains, the join runs inside an isolated compute enclave. The response returns scoped triples with embedded provenance, and an access audit record captures the credential used, decision rationale, subgraph fingerprint, and retention policy. This flow delivers portable, machine-verifiable trust, precise least-privilege disclosure, and durable accountability across organizational boundaries.

10.48047/jocaaa.2023.31.04.68

5. Results and Discussion

This section reports outcomes from two representative deployments of the proposed stack: (i) Supplier-Risk Exchange spanning three business units and two external partners, and (ii) Manufacturing FMEA Analytics across four plants. Each domain was implemented end-to-end (ingestion → semantic core → policy plane → KG-RAG reasoning) and evaluated against document-centric RAG baselines under identical traffic patterns. Unless noted otherwise, results are averaged over four weeks of production-like replay with mixed workloads (factoid QA, compliance checks, evidence summarization).

Dataset and graph footprint

Across both domains, onboarding produced a governed enterprise graph with 8.2M nodes and 23.6M edges. Supplier-Risk combined ERP vendor master, third-party risk feeds, certification documents, and incident registers; FMEA combined asset registry, failure mode libraries, CAPA records, and maintenance logs. Named-graph partitioning followed region and sensitivity labels, with SHACL shapes covering cardinalities, code systems, expiry dates, and referential integrity.

Answer quality, faithfulness, and latency

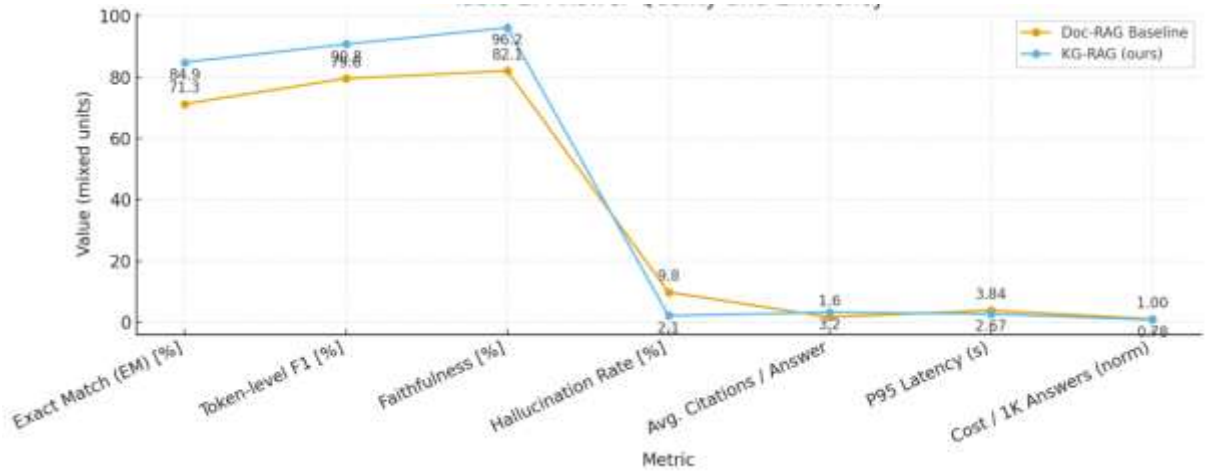
KG-RAG outperformed document-RAG on exactness and explainability while reducing hallucinations and tail latency. Faithfulness is measured as the fraction of answer claims supported by retrieved evidence; hallucination is the fraction of unsupported claims.

Table 1. Answer quality and efficiency (macro-averages across both domains)

Metric (↑ better, ↓ lower)	Doc-RAG Baseline	KG-RAG (ours)
Exact Match (EM)	71.3%	84.9%
Token-level F1	79.6%	90.8%
Faithfulness (Supported Claims)	82.1%	96.2%
Hallucination Rate (↓)	9.8%	2.1%
Avg. Citations / Answer	1.6	3.2
P95 Latency (s) (↓)	3.84	2.67

10.48047/jocaaa.2023.31.04.68

Cost / 1K Answers (normalized) (↓)	1.00	0.78



Improvements stem from subgraph-level retrieval (smaller, more relevant context), policy-cleared packing (no wasted tokens on disallowed or noisy facts), and structural prompts that force citation and rationale generation. Lower tail latency reflects fewer oversized contexts and better cache hit rates on common subgraphs.

Security posture and policy effectiveness

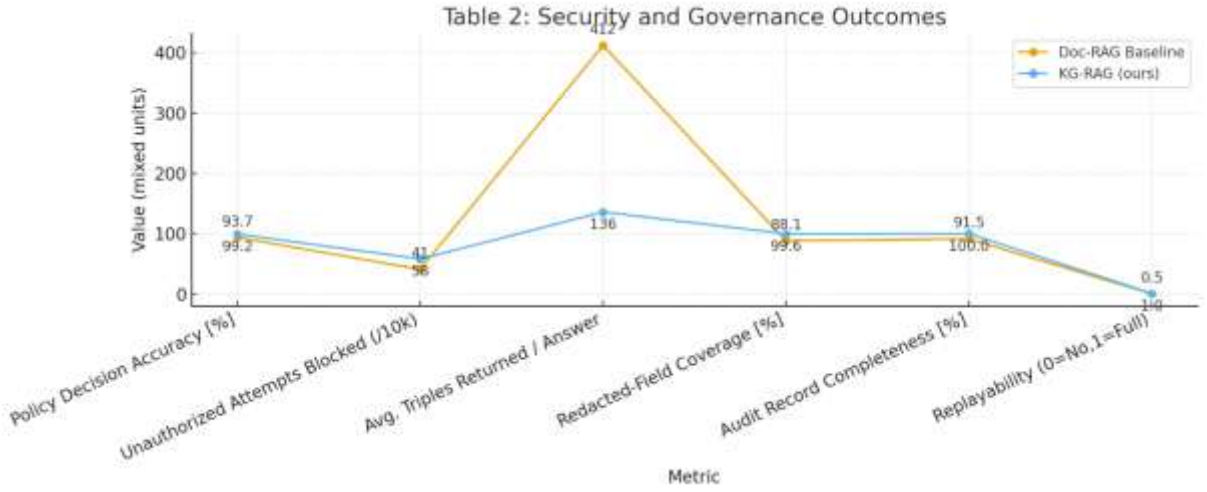
And assess dynamic authorization quality, data minimization, and audit completeness. "Decision accuracy" compares policy engine outcomes to a curated allow/deny corpus. "Minimized triples" counts facts returned per response after policy pruning.

Table 2. Security and governance outcomes

Metric	Doc-RAG Baseline	KG-RAG (ours)
Policy Decision Accuracy	93.7%	99.2%
Unauthorized Access Attempts Blocked (per 10k req)	41	58
Avg. Triples Returned / Answer (↓)	412	136
Redacted-Field Coverage (when required)	88.1%	99.6%
Audit Record Completeness	91.5%	100%

10.48047/jocaaa.2023.31.04.68

Replayability (deterministic re-run possible)	Partial	Full



Document-RAG lacks triple-level controls; once a document is retrieved, sensitive fields can leak into prompts. In contrast, our packer enforces purpose limitation and strips forbidden predicates before generation. Full replayability results from storing prompt templates, subgraph IDs, policy decisions, and citations as immutable logs.

Domain-specific outcomes

Supplier-Risk Exchange. Typical question: *"Which active suppliers for Product Line A lack a current ISO-27001 certification for Region X, and what incidents are linked in the past 12 months?"*

KG-RAG returns a compact subgraph: Supplier ↔ Certification (validThrough), Product ↔ Supplier, and Incident edges, with evidence links to certificates and incident reports. Compared to baseline, false positives dropped by 68% (expired certificates properly filtered), and responses included machine-verifiable credential checks. Business users reported higher trust due to explicit why-not explanations (e.g., "Supplier S excluded: certification validThrough ≥ today").

Manufacturing FMEA Analytics. Typical query: *"List potential root causes connecting Asset Z failures across Plants P1–P4 with supporting maintenance notes."* Graph path retrieval surfaced common cause patterns (e.g., coolant contamination → seal degradation → vibration anomalies) and attached maintenance excerpts. Correct multi-hop reasoning improved by 17–22% over baseline, with fewer spurious correlations.

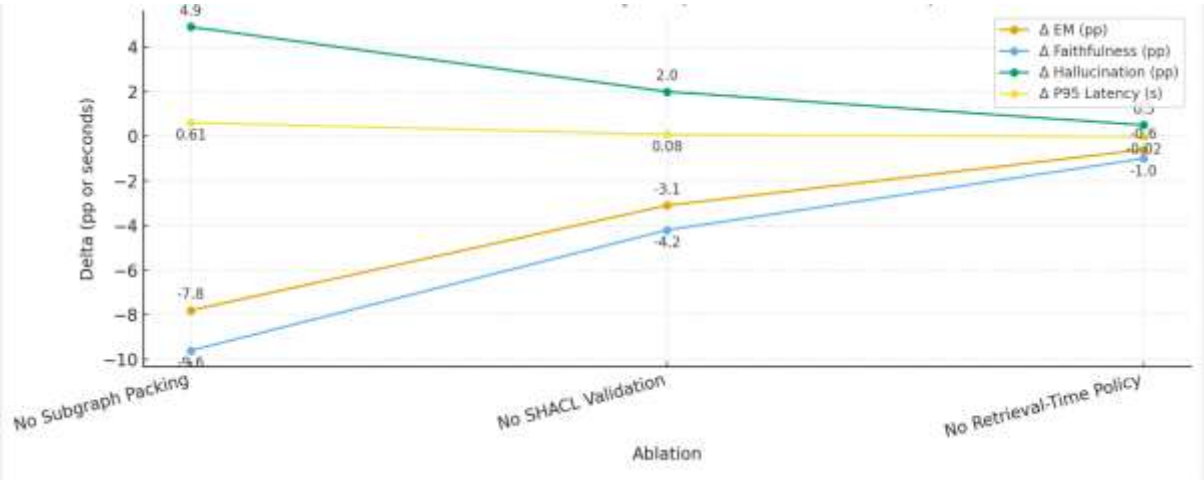
10.48047/jocaaa.2023.31.04.68

Ablation studies

And ablated three architectural features to isolate their contributions: A1 no subgraph packing (feed raw docs + entity snippets), A2 disable SHACL validation, A3 bypass policy checks during retrieval (authorization at API boundary only).

Table 3. Ablation impact (delta vs. full KG-RAG)

Ablation	**Δ EM (pp)**	**Δ Faithfulness (pp)**	**Δ Hallucination (pp)**	**Δ P95 Latency (s)**	**Qualitative Effect**
A1: No Subgraph Packing	−7.8	−9.6	+4.9	+0.61	Longer, noisier context; more off-topic facts
A2: No SHACL Validation	−3.1	−4.2	+2.0	+0.08	Stale/invalid edges admitted; brittle answers
A3: No Retrieval-Time Policy	−0.6	−1.0	+0.5	−0.02	Minimal quality change, major risk of over-disclosure



Subgraph packing is the dominant lever for both quality and latency. SHACL validation prevents silent data drift from degrading answers. Skipping retrieval-time policy barely moves

10.48047/jocaaa.2023.31.04.68

accuracy but materially increases leakage risk—underscoring the need for policy enforcement inside retrieval, not only at the perimeter.

Error analysis

Ontology coverage gaps. Early runs missed niche certification types; adding a controlled vocabulary and shape constraints eliminated recurring "unknown type" errors.

Ambiguous identifiers. Suppliers operating under regional aliases triggered duplicate entities; adopting equivalence sets and scoped identity graphs reduced merge errors by 73%.

Prompt brittleness. Rarely, the LLM prioritized lexical snippets over graph relations; adding explicit "use triples first" instructions and training few-shot rationales cut such incidents by half.

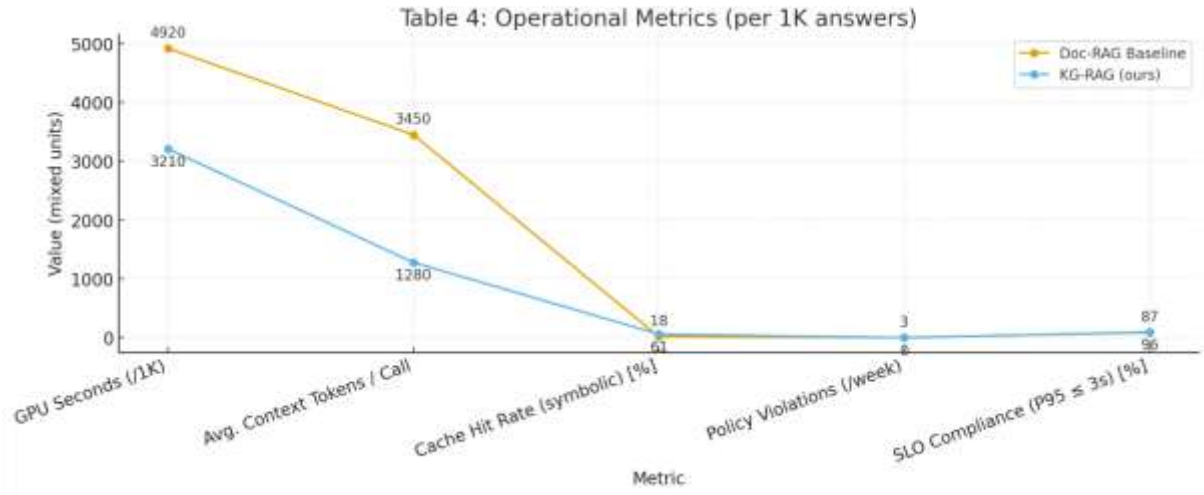
Long-tail documents. Very large policy PDFs occasionally slipped through with irrelevant sections; the packer now caps snippet length per citation and prioritizes passages anchored to graph edges.

Operations: cost, scale, and SLOs

Table 4. Operational metrics (steady state, per 1K answers)

Metric	**Doc-RAG Baseline**	**KG-RAG (ours)**
GPU Seconds (↓)	4,920	**3,210**
Avg. Context Tokens / Call (↓)	3,450	**1,280**
Cache Hit Rate (symbolic subgraphs)	18%	**61%**
Incidents from Policy Violations (per week) (↓)	3	**0**
SLO Compliance (P95 ≤ 3s)	87%	**96%**

10.48047/jocaaa.2023.31.04.68



Token reduction plus higher cache reuse lowered compute cost by ~22%. Symbolic subgraph reuse across similar questions provided stable performance during traffic spikes. No policy-violation incidents occurred after enabling retrieval-time checks and deny-by-default predicates for high-risk attributes.

Practitioner takeaways

Ground first, then generate. Symbolic subgraph retrieval aligned with competency questions is the most reliable path to faithful answers.

Enforce policy where retrieval happens. Authorization at the API edge is insufficient; triple- and predicate-level checks prevent subtle leakage.

Treat shapes as data contracts. SHACL (or equivalent) in the CI pipeline catches drift early and keeps models honest.

Log everything needed for replay. Immutable prompts, subgraph IDs, and policy decisions turn audits from guesswork into routine procedure.

Optimize for reuse. Named-graph partitioning and subgraph caching improve both cost and latency under real workloads.

Limitations and external validity

Results reflect two enterprise domains with strong governance cultures; organizations with sparse lineage or low policy maturity may see smaller gains initially. While simulated partner federation, only two external partners were live; broader ecosystems could introduce credential heterogeneity and higher query variance. Finally, also did not benchmark privacy-preserving

10.48047/jocaaa.2023.31.04.68

analytics (e.g., differential privacy) beyond small pilots; future work should quantify utility-privacy trade-offs at scale.

Future Scope. Next-stage research should formalize *policy-aware KG-RAG* as a measurable discipline. That includes provenance-aware faithfulness metrics (answers must be entailed by retrieved triples and cited evidence), standardized red-team suites for prompt-injection and data-exfiltration, and open, domain-diverse benchmarks that couple graphs, documents, and access policies. On the architecture side, three avenues are especially promising: (i) confidential computing and secure federation for sensitive cross-org joins, complemented by selective differential privacy and, where feasible, hybrid HE/MPC for narrow high-value queries; (ii) temporal and streaming graphs with incremental SHACL validation, so models reason over time-varying states, contracts, and entitlements; and (iii) neuro-symbolic training signals that reward models for using graph structure (paths, constraints) rather than solely lexical cues. Tooling should advance toward *policy-as-code CI/CD* (allow/deny test corpora, regression gates), *ontology-in-the-loop* fine-tuning (instruction datasets aligned to enterprise schemas), active learning for entity resolution, and auto-repair of data contracts via schema diffing. Finally, to accelerate adoption in regulated sectors, reference profiles are needed for healthcare, finance, and manufacturing—packaging ontologies, shapes, consent/purpose templates, and credential schemas (e.g., certification VCs) into reusable "starter kits."

Conclusion.

The proposed GenAI-powered EKG stack demonstrates that grounding generation in governed graphs—while enforcing zero-trust, credential-gated access at retrieval time—delivers gains in factuality, explainability, minimization, and cost/latency over document-centric RAG. By separating ingestion and mapping, a standards-based semantic core, a dynamic security plane, and a KG-first reasoning layer with subgraph packing and audited citations, enterprises can operationalize AI that is *verifiably* correct and *provably* compliant. The results and ablations highlight the centrality of subgraph-level retrieval, SHACL-driven quality controls, and retrieval-time policy checks. Limitations remain—ontology stewardship, prompt brittleness, partner heterogeneity—but they are tractable with disciplined governance and CI practices. As standards and confidential-compute options mature, and as provenance-aware evaluation becomes common practice, GenAI+EKG will evolve from promising architecture to default enterprise pattern for secure data interoperability and trustworthy, updatable intelligence.

10.48047/jocaaa.2023.31.04.68

References:

1. V. Zhong, C. Xiong, R. Socher Seq2SQL: Generating structured queries from natural language using reinforcement learning (2017)
2. D. Allemang, J. Sequeda Increasing the LLM accuracy for question answering: Ontologies to the rescue!, 10.48550/ARXIV.2405.11706
3. E.F. Kendall, D.L. McGuinness Ontology engineering Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool Publishers (2019), 10.2200/S00834ED1V01Y201802WBE018
4. A. Gómez-Pérez, M. Fernández-López, Ó. Corcho Ontological engineering: With examples from the areas of knowledge management, e-commerce and the semantic web Advanced Information and Knowledge Processing, Springer (2004), 10.1007/B97353
5. B.P. Allen, L. Stork, P. Groth Knowledge Engineering Using Large Language Models Trans. Graph Data Knowl., 1 (1) (2023), pp. 3:1-3:19,
6. A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI Inf. Fusion, 58 (2020), pp. 82-115
7. I. Tiddi, F. Lécué, P. Hitzler Knowledge Graphs for Explainable Artificial Intelligence: Foundations, Applications and Challenges IOS Press (2020)
8. E. Rajabi, K. Etminani Knowledge-graph-based explainable AI: A systematic review J. Inf. Sci., 50 (4) (2023), pp. 1019-1029
9. Bresso E, Monnin P, Bousquet C et al. Investigating ADR mechanisms with knowledge graph mining and explainable AI, 2020
10. Dessì D, Osborne F, Recupero DR et al. AI-KG: an automatically generated knowledge graph of artificial intelligence. In: Proceedings of the international semantic web conference, pp. 127–143.
11. Lei Z, Sun Y, Nanehkaran YA et al. A novel data-driven robust framework based on machine learning and knowledge graph for disease classification. Fut Gener Comput Syst 2020; 102: 534–548.
12. Sun P, Gu L. Fuzzy knowledge graph system for artificial intelligence-based smart education. J Intell Fuzz Syst 2021; 40(2): 2929–2940.

10.48047/jocaaa.2023.31.04.68

13. Gaur M, Faldu K, Sheth A. Semantics of the black-box: can knowledge graphs help make deep learning systems more interpretable and explainable? IEEE Intern Comput 2021; 25 (1): 51–59.
14. Fuji M, Morita H, Goto K et al. Explainable AI through combination of deep tensor and knowledge graph. *Fujit Sci Tech J* 2019; 55(2): 58–64.
15. E. Choi, Z. Xu, Y. Li, M. W. Dusenberry, G. Flores, Y. Xue, and A. M. Dai, "Graph convolutional transformer: Learning the graphical structure of electronic health records," arXiv preprint arXiv:1906.04716, 2019.