

## **THE EVOLUTION OF THE SEMANTIC WEB: FROM FOUNDATIONAL TRIPLES TO THE AGENTIC ERA**

Adam ALTĂR – SAMUEL<sup>1</sup>

Daniela Alexandra CRIȘAN<sup>2</sup>

Justina Lavinia STĂNICĂ<sup>3</sup>

### **Abstract**

This paper analyzes the evolution of the Semantic Web into the "Agentic Web", a digital ecosystem designed for autonomous AI agents. This transformation is driven by advanced W3C standards like RDF 1.2, which introduces triple terms to streamline complex metadata management and optimize graph storage. A critical focus of the research is the neuro-symbolic integration of these structured knowledge graphs with Large Language Models (LLMs) to mitigate AI hallucinations and enable accurate reasoning. Additionally, the paper explores how the emerging Web Model Context Protocol allows traditional websites to expose structured, agent-ready tools natively in the browser. Finally, it examines the necessity of Decentralized Identifiers and Verifiable Credentials, which provide a cryptographic trust layer enabling AI agents to securely authenticate, negotiate, and transact across different domains without requiring constant human oversight.

**Keywords:** Semantic Web, Agentic AI, Knowledge Graphs, Large Language Models (LLMs), Web Model Context Protocol (WebMCP)

**JEL Classification:** L86

### **1. Introduction**

The conceptualization of the World Wide Web fundamentally transformed global communication by interlinking documents to facilitate human consumption and knowledge sharing. However, the sheer volume, rapid expansion, and unstructured nature of these digital documents necessitated the development of a parallel architecture—one designed explicitly for machine interpretation rather than human reading. This parallel architecture,

---

<sup>1</sup> PhD, Associate Professor; Department of Computer Science, Mathematics and Statistics; Romanian – American University, Bucharest, Romania; adam.altar@rau.ro

<sup>2</sup> PhD, Associate Professor; Department of Computer Science, Mathematics and Statistics; Romanian – American University, Bucharest, Romania; daniela.alexandra.crisan@rau.ro

<sup>3</sup> PhD, Lecturer; Department of Computer Science, Mathematics and Statistics; Romanian – American University, Bucharest, Romania; lavinia.stanica@rau.ro

known historically and presently as the Semantic Web, was introduced to endow digital information with well-defined meaning, enabling computers and software agents to process, integrate, search, and act upon data autonomously across vast networks [1].

In its earliest iterations, the Semantic Web was viewed primarily as an engineering solution constructed utilizing Uniform Resource Identifier (URI) syntax and the Extensible Markup Language (XML) to standardize the transmission of machine-understandable data [1]. By the year 2026, this academic framework has matured into the critical, structural backbone powering global artificial intelligence systems. The global semantic knowledge graphing market has reached an estimated valuation of \$4.9 billion in 2026, with financial projections indicating a surge to \$15.2 billion by the year 2033, driven by a compound annual growth rate (CAGR) of 17.6%. Furthermore, the broader semantic web market, encompassing data integration tools and inference engines, is projected to reach \$7.73 billion by 2030, scaling at a CAGR of 23.3%.

This rapid expansion is primarily catalyzed by the critical necessity for structured semantic data to support the contextual understanding required by modern Large Language Models (LLMs) and autonomous AI agents [2]. The contemporary Semantic Web is no longer merely a theoretical complement to the existing web of HTML documents; it is the fundamental infrastructure of the "Agentic Web"—an emerging digital ecosystem where intelligent, autonomous software agents handle complex tasks, orchestrate multi-step workflows, negotiate with external services, and transact securely on behalf of human users and enterprise organizations [1].

## **2. The Evolution of the Foundational Data Model: RDF 1.2 and Triple Terms**

The Resource Description Framework (RDF) has long served as the standard data model for interchange on the Web. The core utility of RDF lies in its ability to facilitate data merging and integration even when the underlying schemas of disparate databases differ significantly [1]. Historically, RDF extended the standard linking structure of the Web by utilizing URIs to name the relationships between entities, thereby forming a directed, labeled graph [1]. Within this graph, the edges represent the named links, and the nodes represent the resources. The fundamental unit of this model is the "triple," consisting of a subject, a predicate, and an object [1].

While the triple model provided a mathematically elegant way to represent structured and semi-structured data, the limitations of the RDF 1.1 specification became increasingly apparent as enterprise knowledge graphs grew more complex. Specifically, data architects encountered severe structural inefficiencies when attempting to express metadata about the data itself—such as the provenance of a statement, certainty scores, temporal restrictions, or the specific agent that generated a fact [3]. Under the RDF 1.1 standard, the process of "reification"—expressing an abstract construct with existing concrete methods—required stating four additional triples just to refer to the single target triple for which metadata was

being provided [3]. The subject of these four additional triples required a new identifier, such as a blank node or a new Internationalized Resource Identifier (IRI).

This standard reification approach, while compliant with early specifications, created massive inefficiencies related to persisting and exchanging RDF data [3]. A database containing a billion facts could easily balloon to five billion triples simply by adding basic provenance metadata. Alternative modeling workarounds emerged, such as singleton properties, named graphs, and N-ary relations, but these approaches either increased ontology complexity to unsustainable levels or severely degraded querying efficiency by requiring complex regular expression parsing [4].

In 2026, the World Wide Web Consortium (W3C) successfully addressed these historical structural limitations through the ratification and deployment of the comprehensive RDF 1.2 specifications [5]. The most transformative and widely anticipated feature of the RDF 1.2 data model is the formal integration of "triple terms," a concept heavily incubated in earlier years under the moniker RDF-star [5]. This advancement introduces a fourth kind of RDF term to the abstract syntax, seamlessly allowing an entire RDF triple to be utilized as the subject or the object of another triple [5]. By doing so, the framework natively enables systems to make statements about other statements without the bloat of legacy reification.

The update to the RDF 1.2 abstract data model is accompanied by critical revisions to the concrete serialization formats utilized by developers. The RDF 1.2 Turtle specification introduces highly refined syntactic sugar to make data modeling exponentially more expressive and concise [6]. The specification formally standardizes the `<<(s?p?o)>>` syntax for explicitly defining quoted triple terms [6]. More importantly, it introduces a dedicated annotation syntax, represented by `{... }`, which allows triple terms to be concurrently asserted and annotated [6]. This allows data modelers to state a fact and immediately attach metadata to it within a single, highly readable block of code.

Beyond reification, the W3C RDF & SPARQL Working Group introduced several other vital enhancements. RDF 1.2 introduces directional language-tagged strings, establishing a base direction component that dictates the initial text direction for proper presentation by global user agents [5]. To manage the interoperability transition between legacy enterprise data and modern infrastructures, RDF 1.2 documents now utilize a formal version announcement mechanism, prominently featuring the `VERSION "1.2"` directive at the head of serialization files [6]. This ensures backward compatibility, defining precise syntax conformance boundaries so that data conforming to version "1.1" remains valid as data with version "1.2-basic" [5].

### **3. Advanced Query Mechanics: SPARQL 1.2 and Entailment Regimes**

Because the Semantic Web represents abstract concepts in the form of triplets interconnected within a massive, directed graph, traditional relational database query languages, such as SQL, possess limited and highly inflexible syntaxes for traversing this

data [1]. The SPARQL Protocol and RDF Query Language emerged to solve this problem, acting as the fundamental mechanism for retrieving and manipulating data stored in the Resource Description Framework format [1].

A mathematically sound RDF query language must adhere to five distinct properties to function effectively at enterprise scale: expressiveness (providing relational completeness), closure (ensuring operations on a graph data model return graph data models), adequacy (utilizing all underlying concepts of the data model), orthogonality (ensuring operations are independent of their usage context), and safety (guaranteeing that syntactically correct queries return a finite set of results from finite data sets) [7].

To parallel the structural evolution of the RDF 1.2 data model, the W3C formalized the SPARQL 1.2 specification. SPARQL 1.2 expands upon the foundational SELECT, ASK, DESCRIBE, and CONSTRUCT query forms established in previous versions [1]. The primary directive of SPARQL 1.2 is to safely integrate the querying of the newly introduced triple terms while strictly maintaining the mathematical closure and safety of the graph data model.

To achieve this, the SPARQL 1.2 grammar introduces a suite of specific functions dedicated to manipulating, validating, and extracting components from triple terms. These built-in query functions include TRIPLE, isTRIPLE, SUBJECT, PREDICATE, and OBJECT. This programmatic access allows query execution engines to dissect nested metadata statements natively. In previous architectures, extracting the subject of a reified node required cumbersome graph traversal workarounds; in 2026, a developer can simply query the nested object directly. Additionally, the query language incorporates specific functions related to the new language tags and base directions, namely LANGDIR, hasLANGDIR, and STRLANGDIR, ensuring full coverage of the RDF 1.2 data model.

Furthermore, SPARQL 1.2 significantly updates the Entailment Regimes. Entailment rules are fundamental to the Semantic Web, as they state which implicit information can be logically inferred from explicit information present in the graph [1]. The SPARQL 1.2 Entailment Regimes specification ensures that basic graph pattern matching incorporates standard semantic web entailment relations without breaking the query engine. These regimes include RDF entailment, RDFS entailment, D-entailment, OWL Direct Semantics, and RIF Core entailment. This capability is absolutely critical for deep reasoning engines that must distinguish implicit logical taxonomies from explicit data declarations in real-time, allowing an AI agent to deduce that if an entity is defined as a "Pediatrician," it implicitly belongs to the broader class of "Physician," even if that specific triple is absent from the database.

Federated queries, historically orchestrated via the SERVICE keyword to merge data distributed across the Web, have also been refined. The SPARQL 1.2 Federated Query extension is designed to handle distributed query execution involving the new triple terms and updated entailments, ensuring that linked open data endpoints

can communicate complex metadata structures securely and efficiently. Software implementations, such as the open-source Oxigraph, have rapidly adopted these standards, offering stable support for SPARQL 1.2 updates, experimental operators like LATERAL joins, and advanced property path matching [8].

#### **4. Bridging Relational Silos: The Modern RDB2RDF Mapping Ecosystem**

Despite the inherent architectural advantages of native graph storage and semantic triples, the overwhelming majority of global enterprise data continues to reside in traditional Relational Database Management Systems (RDBMS). Exposing this highly structured, tabular data to the Semantic Web requires an extensive conversion process known as RDB-to-RDF mapping [1]. This process bridges the conceptual gap between relational tables (featuring primary keys, foreign keys, and rows) and the directed graphs of the Semantic Web, allowing legacy data to be seamlessly queried via SPARQL [1].

In the early developmental stages of the Semantic Web, data integration relied on rudimentary tools. Direct mapping was often executed through platforms like Virtuoso, the D2RQ Platform, and Spyder [1]. The D2RQ Platform, for example, offered a system for accessing relational databases as virtual, read-only RDF graphs without replicating the data into a dedicated RDF store, translating SPARQL queries into SQL on the fly. Spyder served as a free, less business-oriented alternative focusing exclusively on direct mapping via early W3C standards [1].

However, by 2026, the mapping ecosystem has undergone a massive, high-performance architectural shift, rendering legacy tools like Spyder and the original D2RQ framework effectively obsolete. The industry has definitively standardized around the W3C RDB2RDF Mapping Language (R2RML) and its highly flexible extension, the RDF Mapping Language (RML). RML extends the capabilities of R2RML to accommodate heterogeneous data sources beyond relational databases, including CSV, JSON, and XML documents.

The contemporary landscape of data mapping is bifurcated into two primary architectural paradigms: Virtual Knowledge Graphs (VKG) and Materialized Knowledge Graphs.

The Virtual Knowledge Graph paradigm is championed predominantly by systems like Ontop. Ontop provides a semantic abstraction layer that entirely hides complex, underlying SQL join logic behind a pristine conceptual ontology. End-users and AI agents interact with the conceptual model using SPARQL, while Ontop dynamically translates these semantic queries into highly optimized SQL, leveraging the underlying relational database's existing execution engine without ever duplicating the data. This is particularly vital for organizations with strict data residency requirements.

Conversely, for architectures requiring deep, multi-hop graph analytics, machine learning training, or high-speed LLM Retrieval-Augmented Generation (RAG) pipelines, materialized knowledge graph construction is preferred. In this space, highly scalable

mapping engines like SDM-RDFizer and Morph-KGC dominate the 2026 landscape [9]. These engines are engineered to execute RML rules to generate high-quality linked data at unprecedented scales. SDM-RDFizer addresses the immense memory storage problems of graph creation by introducing TriplesMap Planning (TMP) to define optimized execution evaluation plans, alongside novel operators for compressing data structures in memory [10]. Morph-KGC similarly achieves scale through sophisticated mapping partitions [9].

Once these massive graphs are physically materialized via ETL pipelines, they are typically hosted in enterprise-grade native cloud platforms. Amazon Neptune, for example, has become a foundational infrastructure piece, seamlessly supporting both property graph models and RDF graphs within the same managed instance, providing automated backups, point-in-time recovery, and deep integration with cloud security identity frameworks [11].

## **5. The Symbiosis of Ontologies and Large Language Models**

Ontologies remain the conceptual bedrock of the Semantic Web. Defined philosophically as the metaphysical study of the nature of being and existence, in digital ecosystems, an ontology represents a formal file that defines relationships, classes, taxonomies, and strict rules of inference between terms within a specific domain [1]. By setting properties for specific overarching classes—with subclasses subsequently inheriting those properties—and applying inference rules, software applications can definitively deduce relationships that are not explicitly stated within the database [1]. The Protégé application, historically recognized as the most flexible ontology editor incorporating Semantic Web requirements [1], has evolved heavily into WebProtégé, integrating directly with the Ontology-Development Kit (ODK) to provide automated support, testing, and continuous integration via GitHub pipelines.

However, the utility and visibility of ontologies have expanded dramatically in 2026 due to the absolute ubiquity of Large Language Models (LLMs). LLMs, built upon advanced Transformer neural network architectures, excel at probabilistic word prediction, contextual language generation, and long-form reasoning based on the unsupervised ingestion of massive pretraining datasets. Yet, despite their linguistic fluency, isolated LLMs fundamentally lack deterministic mechanisms for factual verification, inevitably leading to "hallucinations" and severe logical inconsistencies [12].

The critical integration of Semantic Web technologies with LLMs forms a hybrid, neuro-symbolic architecture that defines modern AI. Knowledge graphs provide the definitive truth and transparency layer that probabilistic language models natively lack. By strictly grounding LLMs in highly curated, ontology-driven knowledge graphs—a sophisticated enterprise process currently known as Context Engineering or GraphRAG—organizations generate conversational AI interfaces that are structurally bound to factual, proprietary data.

This symbiosis forms a bidirectional relationship that effectively bridges the gap between structured databases and unstructured content. On one side, autonomous AI agents are

continuously deployed to ingest raw, legacy text documents, images, and audio, utilizing advanced natural language processing to automatically extract entities and relationships. The AI converts this unstructured data into perfectly formatted RDF triples, effectively automating the continuous population and maintenance of the knowledge graph.

On the other side, the semantic layer allows LLMs to engage in far more complex, multi-hop reasoning, providing highly explainable, plain-language answers to end-users. Because the semantic layer (often utilizing Virtual Knowledge Graphs) maintains strict access controls and data entitlements, the AI can securely retrieve customized answers based on the user's specific organizational role. Open structured data repositories, most notably Wikidata, serve as massive, collaboratively maintained baseline ontologies that provide high-quality training and reference data for these global AI systems, reflecting global diversity and factual transparency.

## **6. Semantic Search: Beyond Keyword Matching and the "Zero-Click" Paradigm**

The maturation of semantic semantics and knowledge graphs has completely restructured the digital search landscape. Historically, traditional search engines relied almost exclusively on keyword density matching and reverse-indexing HTML documents. This approach was highly inefficient, often returning massive volumes of irrelevant pages because the system lacked any contextual awareness of the query's actual meaning [1]. Semantic search, conversely, focuses on comprehending the deeper intent, contextual meaning, and entity relationships behind a user's query [13].

By 2026, semantic search definitively represents the leading application type within the semantic knowledge graphing market, driven heavily by adoption rates scaling rapidly to address contextual results. Instead of matching literal character strings, semantic search engines utilize advanced NLP, vector embeddings, and entity mapping to understand the relationship between words and broader concepts. For example, an enterprise query regarding "hardware for graphic design" will retrieve entities semantically mapped to high RAM constraints and specific graphical processing capabilities, rather than blindly returning pages that simply contain the words "hardware" and "design" [13].

In complex enterprise environments, semantic search platforms seamlessly map conceptual relationships across highly dispersed data repositories (ranging from legacy codebases to HR policies), ensuring that results are highly tailored to the specific needs and permissions of the user [14].

On the public web, this technological evolution has led to an existential paradigm shift in Search Engine Optimization (SEO) and digital marketing. The traditional "10 blue links" format that defined the early web has been largely superseded by AI-driven Answer Engines [14]. This has precipitated the critical "Zero-Click" tipping point—a macroscopic reality where a vast majority of user queries are satisfied instantly by an AI synthesis of semantic

data directly on the search results page, without the user ever requiring a click-through to a third-party website [15].

Consequently, digital visibility and website growth in 2026 rely significantly less on legacy keyword density strategies and almost entirely on the robust deployment of structured schema markup and JSON-LD. By providing machine-readable, RDF-compliant semantic contracts, organizations ensure that their data, product inventory, and services are legible and highly ranked by AI crawlers and autonomous answer engines [16].

## **7. The Evolution of Web Services: From SOAP to the Agentic Protocols**

In the early architectural designs of the Semantic Web, traditional Web services provided the primary methods for software programs to exchange data and invoke remote operations across distributed networks. The absolute dominant protocols of that era were the Simple Object Access Protocol (SOAP) for messaging, the Web Services Description Language (WSDL) for precise interface description, and Universal Description, Discovery, and Integration (UDDI) for publishing and maintaining service registries [1].

By 2026, this legacy architecture has been fundamentally overhauled to support modern web velocities. UDDI is entirely obsolete, having been initially replaced by lighter discovery mechanisms and eventually fully superseded by robust, centralized API gateways and service meshes [17]. SOAP and WSDL, while historically valued for their strict, XML-based contract-first security models, have been largely deprecated in favor of agile, lightweight frameworks [17].

The software industry transitioned heavily to Representational State Transfer (REST) APIs, an architectural style that utilizes lightweight JSON formats, standard HTTP verbs (GET, POST, PUT, DELETE), and mandates stateless communication. Concurrently, GraphQL emerged as a highly powerful, query-based API architecture. GraphQL resolved REST's inherent over-fetching issues by allowing client applications to request highly specific, complex data structures via a single endpoint, heavily utilized in front-end-driven or data-heavy mobile applications.

However, traditional REST and GraphQL APIs were fundamentally designed for human developers building applications for human end-users. As the Semantic Web transitioned deeply into the Agentic Web—an ecosystem where autonomous AI agents must execute tasks across disparate digital systems—new, machine-first protocols were urgently required [2].

Initially, AI agents interacted with the web through highly inefficient visual screen scraping or brittle Document Object Model (DOM) manipulation. The agents were forced to guess the functionality of HTML elements designed for human eyes, leading to extreme latency, high token expenditure, and high error rates.

To definitively solve this interoperability crisis, the W3C Web Machine Learning

Community Group introduced the Web Model Context Protocol (WebMCP) in early 2026. WebMCP effectively bridges the final gap between semantic data schemas and executable AI logic. It allows web applications to natively expose client-side JavaScript functions as structured "tools" to AI agents directly through the browser.

Instead of an AI agent attempting to visually parse a complex HTML calendar to book a flight, a website utilizing WebMCP declares a mathematically rigorous, natural-language described schema defining exactly how the flight-booking function operates. The agent can discover, negotiate, and seamlessly invoke these tools instantly, while securely inheriting the human user's session authorization and exact permission boundaries. This protocol definitively shifts the internet from a human-readable visual interface to an Agent-Ready ecosystem, where capabilities are semantically declared and universally actionable by machine consumers.

Furthermore, decentralized technologies like the Hydra ecosystem have reached maturity. Hydra builds entirely upon JSON-LD (JavaScript Object Notation for Linked Data) to create truly self-describing REST APIs. By embedding a semantic vocabulary directly into the API responses, Hydra enables smart clients and autonomous AI agents to discover new capabilities and traverse complex APIs dynamically without requiring hard-coded client logic, fulfilling the original Semantic Web promise of autonomous service discovery and execution.

## **8. The Trust Layer: Decentralized Identifiers and Verifiable Credentials**

A critical, indispensable component of a functional, autonomous Semantic Web is the establishment of verifiable trust. In earlier frameworks, digital signatures—encrypted data packets ensuring a source's authenticity—were the primary mechanism by which software agents verified transactions and guarded against malicious actors [1]. However, centralized certificate authorities and isolated trust registries presented systemic vulnerabilities, friction, and privacy concerns.

By 2026, the absolute necessity for robust, scalable digital trust—driven by heavy regulatory pressures and massive enterprise demand for fraud-proof identity management—has led to the widespread adoption of Decentralized Identity frameworks. The foundation of this new semantic trust layer relies entirely on open W3C standards for Decentralized Identifiers (DIDs) and the Verifiable Credentials (VC) Data Model v2.0 [18].

A Decentralized Identifier (DID) is a globally unique identifier that resolves directly to a DID Document containing public cryptographic keys, verification methods, and service endpoints [18]. Crucially, the DID is entirely controlled by the subject rather than a central issuing authority. Verifiable Credentials are tamper-evident, cryptographically signed digital documents (often packaged natively using JSON-LD for maximum semantic interoperability) that definitively prove specific claims about a person, an organization, or an IoT device.

This paradigm operates on an elegant, mathematically secure triangle of trust: an Issuer cryptographically signs a credential, a Holder stores it locally in a digital wallet, and a Verifier seamlessly checks its authenticity [18]. Because the credentials and DIDs are anchored to immutable decentralized ledgers or distributed trust frameworks, verifying the absolute authenticity of the data requires no real-time integration or API callbacks to the issuing authority. This privacy-by-design architecture eliminates centralized honeypots of personal data and vastly accelerates onboarding [18].

Most importantly, in 2026, this trust layer is no longer limited strictly to human identity; it is the fundamental prerequisite for the Agentic Web [2]. If autonomous AI systems are to execute high-stakes financial transactions, access highly structured enterprise data, and operate safely across disparate organizational boundaries, they must possess identifiable, verifiable authority. Systems can utilize Verifiable Credentials to assign cryptographic, granular identity directly to AI agents [18]. Through decentralized governance files containing machine-readable "trust lists," an AI agent can instantly recognize and verify the delegated authority of another agent, establishing a mathematically provable foundation for machine-to-machine collaboration, negotiation, and data exchange without requiring constant human oversight [2].

## **9. The Agentic Web in Action: Complex Autonomous Scenarios**

To truly understand the practical, real-world realization of the 2026 Semantic Web, it is highly useful to revisit the classic use-case scenario of a semantic agent organizing a holiday to London, as originally conceptualized in the early days of the technology [1]. In the early theoretical vision, a human user would type a rigid sentence ("I want to go to London on Holiday from 8 to 15 June"). The agent would process the text, utilize RDF files and ontologies to recognize the concepts of "London" and "Holiday," invoke separate, disconnected web services for flight and hotel bookings, and return a static list of options for the user to manually review [1].

In the fully realized 2026 Agentic Web, this process is vastly more sophisticated, entirely autonomous, and highly conversational. The user no longer types a rigid command; they engage in a continuous, context-aware conversational interface powered by a multimodal LLM. The core intelligence is no longer a single script, but rather an orchestration of specialized sub-tasks managed by multi-agent frameworks.

When the human user requests the London trip, the primary orchestrating agent parses the intent and instantly accesses the user's personal semantic knowledge graph to retrieve deep historical preferences (e.g., a strong preference for boutique hotels over chains, specific dietary restrictions, past travel stressors, and preferred airlines). The orchestrator then spawns dedicated sub-agents: a flight researcher agent, an accommodation agent, and a budget tracker.

These sub-agents interact directly with the open web. However, rather than clumsily scraping airline HTML websites, the flight agent utilizes the new WebMCP protocol to instantly discover and seamlessly invoke the structured booking functions of various airlines, comparing real-time, RDF-compliant pricing and availability schemas.

If a severe disruption occurs—such as a sudden flight cancellation due to weather—the agentic system does not simply fail and return an error code to the user. It utilizes the deep semantic relationships embedded in the travel ontology to dynamically and proactively reroute the itinerary. The agent interacts autonomously with hotel APIs via JSON-LD payloads to adjust check-in dates and secure alternate transport without the user asking.

When the fully optimized itinerary is compiled, the agent presents it to the user. Upon the user simply stating "Book it", the agent utilizes its Verifiable Credential to cryptographically prove to the airline and hotel that it possesses the delegated financial authority to act on the user's behalf [18]. It executes the secure transaction, standardizes the receipt data into RDF triples, and securely updates the user's personal calendar. Throughout this entire autonomous process, robust Agentic AI defenses are deployed: the agent operates within a Trusted Execution Environment (TEE), utilizes capability-based least privilege to ensure it cannot exceed its authorized budget, and logs every consequential decision using human-readable reasoning chains (CoT) to ensure continuous explainability and auditability.

## **10. Transforming Education: Educational Ontologies and Ambient Learning**

The deep integration of Semantic Web technologies has equally revolutionized the domain of education and e-learning. Historically, the Semantic Web was identified as an exceptionally excellent platform for e-learning because learning materials could be interpreted semantically and reorganized dynamically based on user requests. This process relied heavily on semantic queries and navigation through learning materials made possible by ontologies providing exact, machine-readable definitions of complex concepts [1].

By 2026, the traditional, monolithic Learning Management System (LMS)—which merely hosted and deployed static courses—has been largely supplanted by sophisticated "Ambient Learning" architectures. The psychological reality of the "Forgetting Curve"—the phenomenon where humans forget approximately 70% of new information within 24 hours of exposure—is actively mitigated by integrating education directly into the daily workflow. The enterprise strategy has shifted radically from mandatory, periodic "Time-Out Training" to daily, AI-modularized contextual learning, drastically increasing knowledge retention rates to 80% via automated spaced repetition.

This transformation is heavily reliant on the deployment of advanced educational ontologies and semantic metadata. Ontologies standardize the complex vocabulary between learners, human instructors, and automated learning systems, allowing for a coherent organization of knowledge. These domain ontologies model the entire educational

spectrum, including curricula, skill-based requirements, and pedagogical models. When structured data (people, processes, equipment) and unstructured documents are semantically annotated and bridged via enterprise knowledge graphs, they become unified, highly discoverable "knowledge assets".

Intelligent tutoring systems leverage this semantic architecture to generate deeply personalized, adaptive learning paths. By continuously evaluating the learner's current knowledge state against the target ontology, the system dynamically adapts content. It utilizes the relationships defined within the ontology to infer new knowledge pathways, heavily facilitating discovery learning.

Furthermore, Large Language Models integrated directly with these semantic layers automatically standardize legacy educational content. The AI extracts metadata and cleans massive repositories of knowledge assets in minutes—a task that previously required thousands of human hours. This vast acceleration expedites the onboarding of new staff and the development of in-house talent. Because these educational resources are semantically described using W3C standards like RDF and OWL, the resulting data interoperability ensures that highly personalized learning modules can be seamlessly shared, verified, and reused across different digital educational platforms and virtual environments globally.

## **11. Conclusions**

The trajectory of the Semantic Web over the past decade represents a monumental, structural shift from theoretical data interoperability to the functional, secure execution of global autonomous intelligence. The W3C's maturation of the RDF 1.2 and SPARQL 1.2 specifications fundamentally resolved historical inefficiencies in semantic modeling. The introduction of triple terms and advanced annotation syntaxes provided the mathematically sound, scalable data structures desperately required to manage the complex metadata and nested logic demanded by modern AI systems. Concurrently, the modernization of mapping architectures—shifting from rigid, legacy direct-mapping tools to dynamic Virtual Knowledge Graphs and massively scalable materialized graph databases—has successfully bridged the gap between the world's vast relational data silos and the semantic ecosystem.

However, the true realization of the Semantic Web in 2026 is defined entirely by its convergence with Large Language Models and Artificial Intelligence. The severe limitations of probabilistic LLMs have been mitigated by the deterministic truth provided by enterprise knowledge graphs, creating highly reliable, hybrid neuro-symbolic systems. This foundation has given rise to the Agentic Web, rendering obsolete the traditional methods of human-computer interaction and heavily deprecating legacy web services like SOAP. Through innovative protocols like WebMCP and Hydra, the internet has been aggressively retrofitted for machine consumption, allowing AI agents to seamlessly discover, negotiate, and execute complex workflows directly within the browser. Supported by the immutable security of Decentralized Identifiers and Verifiable Credentials, these autonomous systems operate with mathematically verifiable authority and zero-trust

security. Ultimately, the Semantic Web has fulfilled its original, visionary promise: transforming the internet into a cohesive, intelligent environment where machines do not merely display data, but truly understand, reason, and autonomously act upon it to augment human capability.

## References

- [1] A. ALTĂR-SAMUEL, A. COSTIN, and D. ENACHE, "RDF & RDF Query Languages – building blocks for the semantic web", *Journal of Information Systems & Operations Management*, vol. 9, nr.1, 2015.
- [2] Y. YANG et al., "Agentic Web: Weaving the Next Web with AI Agents", *arXiv preprint arXiv:2507.21206*, July 2025.
- [3] P. WARREN and P. MULHOLLAND, "Evaluating Reification with Multi-Valued Properties in a Knowledge Graph", in *Proc. Knowledge Graphs and Semantic Web Conference (KGSWC)*, Madrid, Spain, pp. 92-105, 2022.
- [4] J. ARENAS-GUERRERO et al., "Declarative generation of RDF-star graphs from heterogeneous data", *Semantic Web*, vol. 16, no. 2, pp. 1-18, 2025.
- [5] G. KELLOGG, O. HARTIG, P.-A. CHAMPIN, and A. SEABORNE, "RDF 1.2 Concepts and Abstract Data Model", *W3C Working Draft*, March 2026.
- [6] G. KELLOGG and D. TOMASZUK, "RDF 1.2 Turtle: Terse RDF Triple Language", *W3C Working Draft*, January 2026.
- [7] P. HASSE, J. BROEKSTRA, A. EBERHART, and R. Volz, "A Comparison of RDF Query Languages", *AIFB Institute*, Univ. of Karlsruhe, Germany, 2007.
- [8] J. DE SMET and R. TAELEMAN, "Traquila: Providing a Foundation for The Evolving SPARQL Ecosystem Through Modular Query Parsing, Transformation, and Generation", in *Proc. 23rd Extended Semantic Web Conference (ESWC)*, 2026.
- [9] J. ARENAS-GUERRERO et al., "Morph-KGC: Scalable knowledge graph materialization with mapping partitions", *Semantic Web*, vol. 15, no. 1, pp. 1-20, 2024.
- [10] E. IGLESIAS, M.-E. VIDAL, D. COLLARANA VARGAS, and D. CHAVES-FRAGA, "Empowering the SDM-RDFizer tool for scaling up to complex knowledge graph creation pipelines", *Semantic Web*, vol. 16, no. 2, 2025.
- [11] W. LI, S. WANG, S. WU, Z. GU, and Y. TIAN, "Performance benchmark on semantic web repositories for spatially explicit knowledge graph applications", *Computers, Environment and Urban Systems*, vol. 95, 2022.

- [12] M. BARONE, A. ROMANO, G. RICCIO, M. POSTIGLIONE, and V. MOSCATO, "Combining evidence and reasoning for biomedical fact-checking", in *Proc. 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1087-1097, 2025.
- [13] M. A. FERRAG, A. LAKAS, N. TIHANYI, and M. DEBBAH, "LLM and AI Agents for Autonomous Systems: A Survey of Applications, Datasets, and Security Challenges", *IEEE Open Journal of Intelligent Transport Systems*, vol. 7, pp. 1-20, February 2026.
- [14] Y. HUANG and G. ZENG, "RD-P: A Trustworthy Retrieval-Augmented Prompter with Knowledge Graphs for LLMs", in *Proc. 33rd ACM Int. Conf. on Information and Knowledge Management*, New York, NY, pp. 942-952, October 2024.
- [15] X. REN, J. TANG, D. YIN, N. CHAWLA, and C. HUANG, "A Survey of Large Language Models for Graphs", in *Proc. 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Barcelona, Spain, pp. 6616-6626, Aug. 2024.
- [16] M. H. DANG, T. H. T. PHAM, P. MOLLI et al., "Llm4schema.org: Generating schema.org markups with large language models", *Semantic Web*, vol. 16, 2025.
- [17] J. SARKAR and G. AGNIHOTRAM, "Model Context Protocol: The Central Nervous System for a New Generation of Artificially Intelligent Agents", *Scientific Journal of Computer Science.*, vol. 1, no. 2, pp. 84-93, December 2025.
- [18] S. ACHEME and G. N. EDEGBE, "A Decentralized Identity Framework for Secure Federated Learning in Healthcare", *Journal of Cyber Security*, vol. 8, no. 1, pp. 1-31, January 2026.

## **Bibliography**

ACHEME S. and EDEGBE G. N., "A Decentralized Identity Framework for Secure Federated Learning in Healthcare", *Journal of Cyber Security*, vol. 8, no. 1, pp. 1-31, January 2026.

ALTĂR-SAMUEL A., COSTIN A., and ENACHE D., "RDF & RDF Query Languages – building blocks for the semantic web", *Journal of Information Systems & Operations Management*, vol. 9, nr.1, 2015.

ARENAS-GUERRERO J. et al., "Declarative generation of RDF-star graphs from heterogeneous data", *Semantic Web*, vol. 16, no. 2, pp. 1-18, 2025.

ARENAS-GUERRERO J. et al., "Morph-KGC: Scalable knowledge graph materialization with mapping partitions", *Semantic Web*, vol. 15, no. 1, pp. 1-20, 2024.

BARONE M., ROMANO A., RICCIO G., POSTIGLIONE M., and MOSCATO V., "Combining evidence and reasoning for biomedical fact-checking", in *Proc. 48th*

*International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1087-1097, 2025.

DANG M. H., PHAM T. H. T., MOLLI P. et al., "Llm4schema.org: Generating schema.org markups with large language models", *Semantic Web*, vol. 16, 2025.

DE SMET J. and TAELMAN R., "Traquila: Providing a Foundation for The Evolving SPARQL Ecosystem Through Modular Query Parsing, Transformation, and Generation", in *Proc. 23rd Extended Semantic Web Conference (ESWC)*, 2026.

FERRAG M. A., LAKAS A., TIHANYI N., and DEBBAH M., "LLM and AI Agents for Autonomous Systems: A Survey of Applications, Datasets, and Security Challenges", *IEEE Open Journal of Intelligent Transport Systems*, vol. 7, pp. 1-20, February 2026.

HASSE P., BROEKSTRA J., EBERHART A., and VOLZ R., "A Comparison of RDF Query Languages", *AIFB Institute*, Univ. of Karlsruhe, Germany, 2007.

HUANG Y. and ZENG G., "RD-P: A Trustworthy Retrieval-Augmented Prompter with Knowledge Graphs for LLMs", in *Proc. 33rd ACM Int. Conf. on Information and Knowledge Management*, New York, NY, pp. 942-952, October 2024.

IGLESIAS E., VIDAL M.-E., COLLARANA VARGAS D., and CHAVES-FRAGA D., "Empowering the SDM-RDFizer tool for scaling up to complex knowledge graph creation pipelines", *Semantic Web*, vol. 16, no. 2, 2025.

KELLOGG G. and TOMASZUK D., "RDF 1.2 Turtle: Terse RDF Triple Language", *W3C Working Draft*, January 2026.

KELLOGG G., HARTIG O., CHAMPIN P.-A., and SEABORNE A., "RDF 1.2 Concepts and Abstract Data Model", *W3C Working Draft*, March 2026.

LI W., WANG S., WU S., GU Z., and TIAN Y., "Performance benchmark on semantic web repositories for spatially explicit knowledge graph applications", *Computers, Environment and Urban Systems*, vol. 95, 2022.

REN X., TANG J., YIN D., CHAWLA N., and HUANG C., "A Survey of Large Language Models for Graphs", in *Proc. 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Barcelona, Spain, pp. 6616-6626, Aug. 2024.

SARKAR J. and AGNIHOTRAM G., "Model Context Protocol: The Central Nervous System for a New Generation of Artificially Intelligent Agents", *Scientific Journal of Computer Science*, vol. 1, no. 2, pp. 84-93, December 2025.

WARREN P. and MULHOLLAND P., "Evaluating Reification with Multi-Valued Properties in a Knowledge Graph", in *Proc. Knowledge Graphs and Semantic Web Conference (KGSWC)*, Madrid, Spain, pp. 92-105, 2022.

YANG Y. et al., "Agentic Web: Weaving the Next Web with AI Agents", *arXiv preprint arXiv:2507.21206*, July 2025.

<https://www.persistencemarketresearch.com/market-research/semantic-knowledge-graphing-market.asp>, Semantic Knowledge Graphing Market Size, Share, and Growth Forecast, 2026 - 2033, Date of last access as 24-02-2026.

<https://www.marketsandmarkets.com/Market-Reports/semantic-web-market-15328110.html>, Semantic Web Market Size, Share, Growth & Forecast 2030 – Markets and Markets, Date of last access as 22-02-2026.

<https://dev.to/vaib/the-agentic-web-ai-agents-and-the-power-of-semantic-data-4o2e>, The Agentic Web: AI Agents and the Power of Semantic Data - DEV Community, Date of last access as 21-02-2026.

<https://www.netcomlearning.com/blog/diving-into-the-era-of-open-agentic-web>, The Open Agentic Web: How AI Agents Are Reshaping Work in 2026 - NetCom Learning, Date of last access as 24-02-2026.

<https://graphdb.ontotext.com/documentation/11.2/rdf-sparql-star.html>, RDF-star and SPARQL-star — GraphDB 11.2 documentation, Date of last access as 20-02-2026.

<https://www.ontotext.com/knowledgehub/fundamentals/what-is-rdf-star/>, What Is RDF-star | Ontotext Fundamentals, Date of last access as 24-02-2026.

<https://www.w3.org/groups/wg/rdf-star/publications>, RDF & SPARQL Working Group - Publications - W3C, Date of last access as 18-02-2026.

<https://w3c.github.io/rdf-new/spec/>, What's New in RDF 1.2, Date of last access as 18-02-2026.