GRAPH RESEARCH LABS

Knowledge graph and agentic AI infrastructure for regulated organisations

About Dougal Watt

Dougal is the co-founder and principal of Graph Research Labs. He is a former IBM Chief Technologist, certified Information Architect at the highest practitioner level, named inventor on multiple patents, and an international speaker on semantic AI and knowledge graph topics. He has spent decades in the real world, designing and building complex IT systems across four continents before starting Graph Research Labs.

Background

There's a recurring pattern across enterprise semantic projects. Ontology teams produce careful, governed models that have nowhere to be tested except in review. Engineering teams build graph infrastructure that's hard to evolve and govern as the model changes. The gap between these two worlds is where most knowledge graph initiatives stall, and it's why semantic IT has been slow to reach mainstream enterprise adoption for two decades.

What's changing is the urgency. LLM hallucinations, AI agents acting without governance, and retrieval systems that return plausible-but-wrong answers have done what decades of semantic-web advocacy could not: made the case for ontology-driven foundations obvious to enterprise leadership. The semantic layer isn't an abstraction anymore, it's the governance and context substrate agentic AI needs to be reliable in production.

Graph Research Labs closes the gap

GRL tooling can integrate with your existing graph and stack, generate production-grade APIs and data products from your ontology in minutes, and ship them into the hands of the developers and AI agents that consume them at the speed your deployment process allows. Semantic infrastructure runs in your environment, not ours. Agents become reliable because the context they reason over is governed. New data products move from concept to production in weeks rather than quarters.

> *The knowledge graph program stops being an aspiration and starts being essential AI infrastructure the business operates on.*

Our set of data and AI tools: GRL Generators, GRL Ontology Manager, GRL Semantic Agent Harness, GRL Governance Metagraph, and more, together with our Strategic Architecture Engagements, takes clients from assessment through to successful production deployment.

Each of those tools exists because the work is genuinely hard. Declarative generation of production systems from ontologies is one of the hardest problems in enterprise semantic infrastructure, and few do it well at production scale. The patterns behind the tooling have compounded over Dougal’s decades

of building and deploying large, complex architectural and information programmes, with patents pending around the core. That depth is what lets GRL multiply knowledge graph teams instead of replacing them, and lets us make your existing graph consumable across the rest of the organisation, including by the agents that need governed APIs to be reliable rather than uncontrolled.

How GRL is Structurally Different

Most credible vendors in the enterprise semantic space are platform plays. You buy their platform; your data and applications live inside their runtime; your governance, generation, and integration happen on their terms. Switching costs are high. Adopting them means committing to their roadmap on their terms.

GRL is different. We don't sell a runtime, we generate standard production data assets that ship into the stack you already have. Four distinctions are worth naming:

01

Stack-agnostic across the semantic graph ecosystem

We work with any SPARQL-compliant graph database you've chosen: Neptune, Stardog, GraphDB, AllegroGraph, Virtuoso, Fluree, RDFox, MarkLogic, your own, instead of asking you to migrate to ours. Your existing investment is protected. GRL is built on the W3C standards stack (RDF, OWL, SPARQL, SHACL), the same ecosystem where industry standards like FIBO (financial services), CDISC (pharma), and GLEIF (regulatory) live. Property-graph-only systems like Neo4j are a different ecosystem optimised for different use cases.

02

Generates data assets your engineers already know how to operate

What GRL produces - APIs, schemas, integration code, agent constraints, are standard production artefacts. Your devops teams maintain them with tools they already know.

03

Runs inside your environment — a security and compliance asset, not a liability

GRL tooling operates in your data centres, your cloud, and your security perimeter, not ours. Customer data never leaves your control to be processed by an external runtime. No third-party SaaS tenant to certify, no vendor employees with access to your data, no outbound data flows to monitor. For regulated environments such as defence, financial services, pharma, or healthcare, this means GRL plugs into the security and compliance posture you've already built, rather than asking you to extend trust to an external system. Data sovereignty, classification, and residency requirements are preserved because GRL never has any access to your data.

04

Ages with your stack, not against it

When the technology landscape shifts — new graph databases, new ontology managers, new AI platforms in the future, GRL data assets adapt. You're not locked into technology that's stuck on its 2026 architecture. Architectural optionality is preserved.

THE BUSINESS TRANSLATION

GRL is designed to fit the existing stack, produce data assets engineering teams already understand, operate inside the security and regulatory perimeter already in place, and leave the resulting assets owned and controlled by the client.

For regulated enterprise — where every investment is measured in risk - an architecturally designed AI semantic product, with audit and provenance built in rather than tacked on, is the difference between an AI initiative and an AI capability.

Beyond Semantic RAG: why governed reasoning beats retrieval

How GRL goes structurally further than GraphRAG and semantic RAG approaches

Semantic RAG (sometimes called GraphRAG) is the sophisticated version of a data retrieval pattern; instead of plain vector search, the agent traverses a knowledge graph to find relevant context. It improves on naive RAG meaningfully, moving production accuracy from the 50–60% range to 70–80% in many deployments. But it hits a ceiling. The structural reason is that semantic RAG still leaves the LLM in charge of what to do with the retrieved graph. The agent retrieves better data but then interprets it freely. Most implementations also use property graphs with ad-hoc schemas that evolve organically, not formally-governed ontologies with type safety, SHACL validation, and standards alignment. The audit trail covers what was retrieved, not what was constrained. Policy enforcement remains post-hoc, hoping the model honours the prompt rather than enforcing the rule structurally.

GRL operates differently. The agent doesn't just retrieve from a graph; it executes against typed reasoning paths defined by a governed ontology using open W3C standards (RDF, OWL, SPARQL, SHACL). This is the same ecosystem where industry standards like FIBO (financial), CDISC (pharma), GLEIF (regulatory), DoDAF / IDEAS / SysML (defence), and Allotrope (lab data) live. SHACL constraints validate every action at runtime; invalid actions are blocked before execution, not detected after. Audit records trace each decision to the specific rule that authorised it.

Production accuracy reaches 90%+ — not because retrieval is better, but because the agent operates inside a structural envelope, defined by your rules, that it cannot escape.

For regulated environments where actions have consequences and approvals require evidence, this is the difference between augmenting workflows and being trusted with production decisions.

TO DISCUSS

To learn more, contact: **dougal.watt@graphresearchlabs.com**.