

Ontology, and how an LLM uses it

THE CONCEPT

01 What is an ontology?

An ontology is a formal model of the key entities in a domain and the **relationships between them**. It defines the vocabulary, structure, and rules that describe how things relate and what they mean.

ENTITIES	RELATIONSHIPS	PROPERTIES
Service	Incident affects Service	Service: name, description, criticality
Incident	Service deployed_to Deployment	Incident: severity, status, created_at
Deployment	Service contains_code Repository	Deployment: environment, version, deployed_at
Team	Service depends_on Team	
Repository		

02 How the LLM uses the ontology



03 Why it matters

Shared meaning Everyone uses the same definitions and relationships.	Better retrieval Finds the right data by understanding context, not just keywords.
Reliable reasoning The LLM's reasoning is grounded in the trusted ontology and real data.	Scalable and governed Evolves with the business through versioning, permissions and controlled changes.

In short

The ontology is the map. The LLM is the navigator. Together they find answers accurately and reliably.

IMPLEMENTATION LAYERS

01 Ontology model Metadata layer

- Defines the schema of the domain
- Object types, properties, relationships
- Actions and constraints
- Permissions and lifecycle status

Metadata store *relational / JSON / Protobuf*

- Versioned
- Governed
- Small in size

02 Data and graph storage Operational layer

- The actual objects and relationships (facts)
- Objects with properties, edges
- Time validity and provenance
- Permissions

Graph / relational store
Postgres, SQL or distributed DB

- Indexed
- ACID
- Source of truth

03 Semantic index Vector and search layer

- Embeds object descriptions, documents and events
- Hybrid search, keyword + vector
- Filtered by type, time and permissions

Vector DB *pgvector, Qdrant, Pinecone*

- Approximate NN search
- Filtered search
- Metadata attached

04 Context building and caching Performance layer

- Builds focused neighborhoods for the LLM
- Expands structural and semantic neighbors
- Applies scoring and limits
- Caches query patterns and contexts

Context cache *Redis / DB / object store*

- TTL based
- Access frequency
- Invalidated on change

05 LLM and application Consumption layer

- Interprets user intent
- Uses the ontology to plan
- Retrieves data and context
- Generates answers with citations

Flow

- Question
- LLM
- Ontology grounding
- Context
- Answer with sources

Cross-cutting concerns

Governance	Versioning	Lineage	Monitoring	Security
Approvals, roles, ownership	Branch, test, migrate, merge	Track data and changes	Freshness, coverage, constraints	Permissions, row and column access

Exactness for truth. Semantics for discovery. Ontology for structure. LLM for understanding.