Ontotext GraphDB

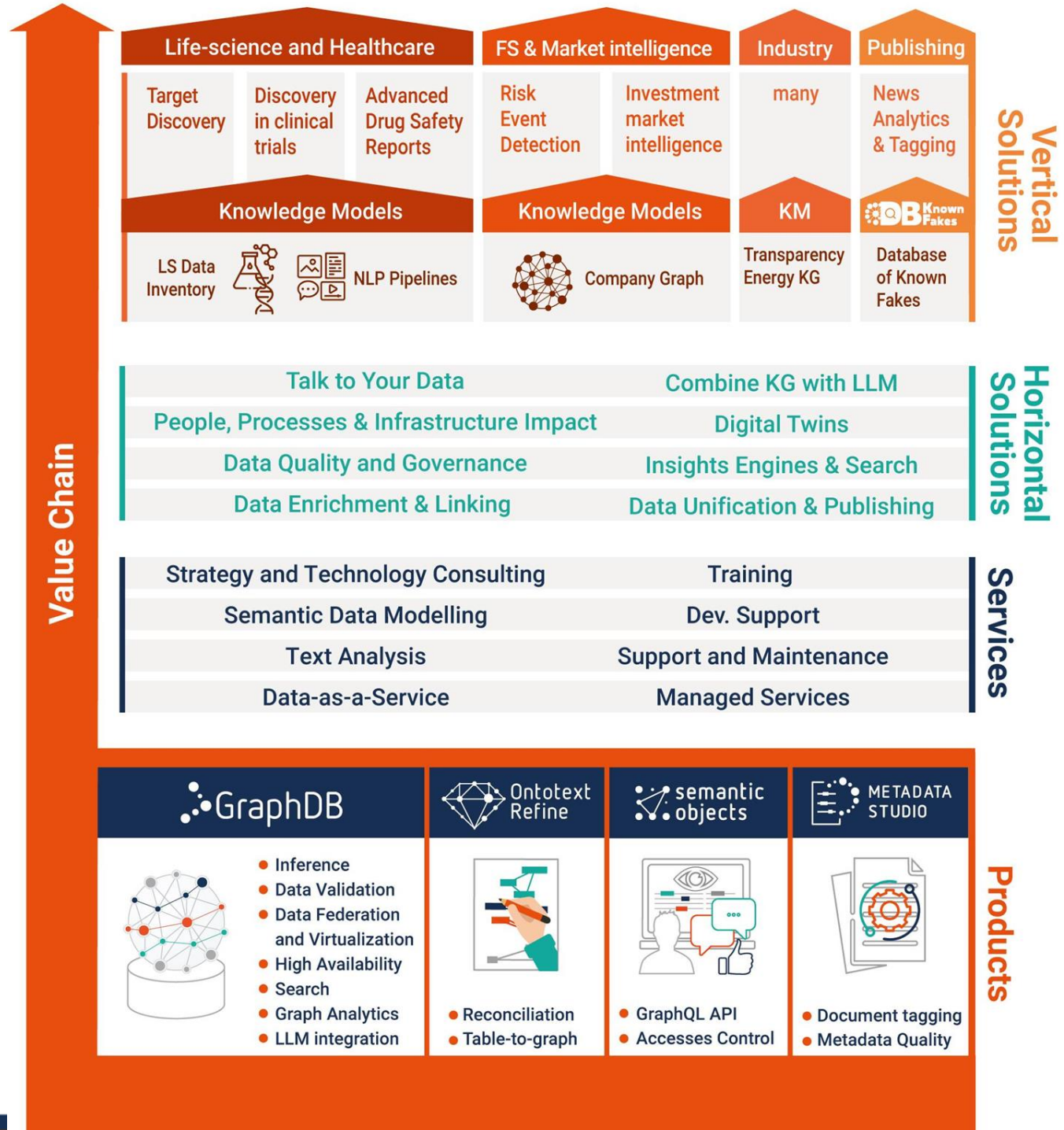
Product Overview

- Intro & Ecosystem
- Essentials
- High Availability & Security
- Workbench & Visualizations
- Search & Analytics
- Reasoning & Validation
- Edge Properties, RDF-star
- Graph Path Search
- Geographic Query Extensions
- Benchmarking
- Latest Releases

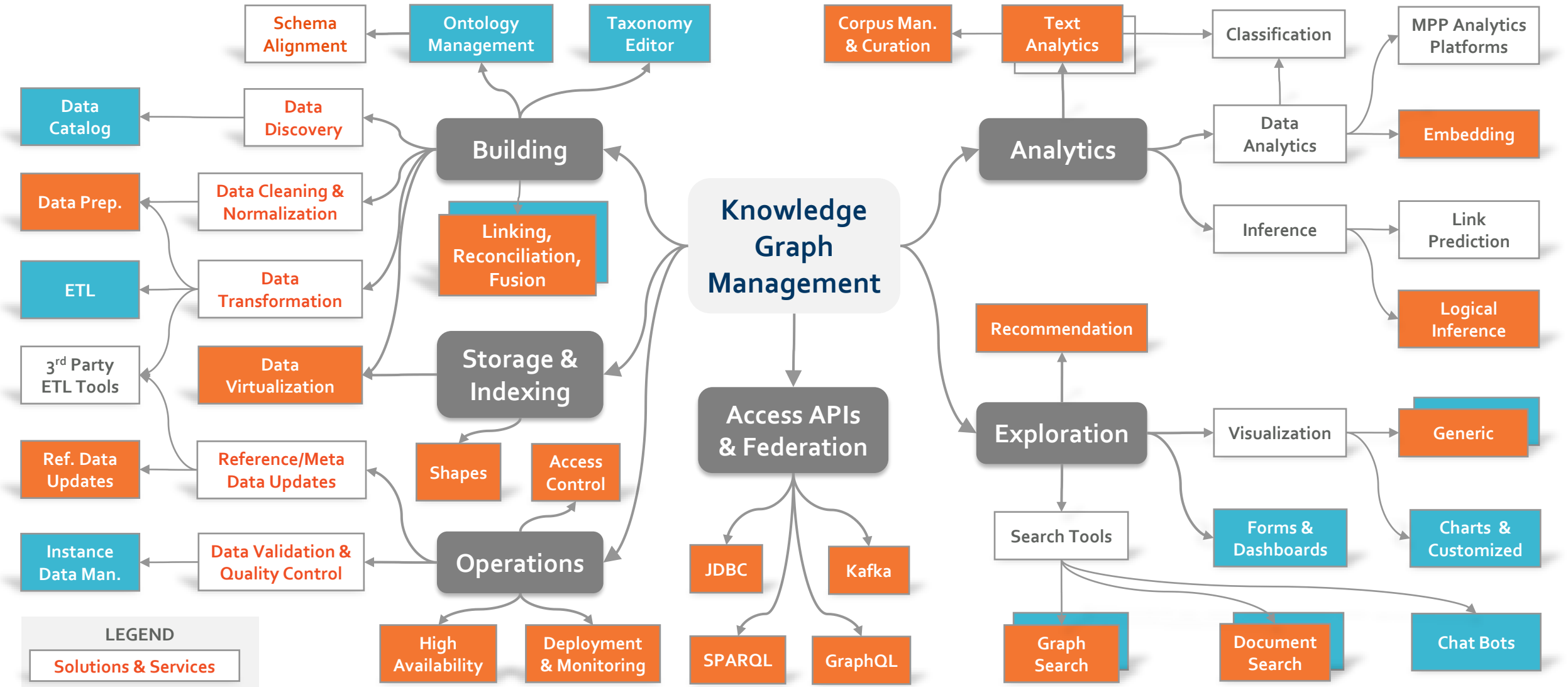
The Best engine for Knowledge Graph Management

Ontotext GraphDB™, and the rich ecosystem of tools on top of it, offer best of bread capabilities to:

- Link and manage diverse data
- Build large knowledge graphs and enrich them via text analysis
- Vast variety of options for search, exploration and analytics



Knowledge Graph Management Capabilities



ontotext

GraphDB™: The Most Versatile Graph Database

- **Dependable performance** across a wide variety of workloads
  - ✓ Efficient handling of both graph analytics and semantic metadata management
    - The only engine passing both LDBC Semantic Publishing and Social Network Benchmarks!
    - Managing 1B facts in 16GB of RAM worth more than loading a trillion triples in a supercomputer
- **Ultimate extensibility, accessibility and deployment options**
  - ✓ Open-source workbench and engine plugins, FTS connectors, GraphQL, SQL/R2RML, JDBC, Kafka
  - ✓ Platform agnostic, Dockerized, portable Java implementation, available at AWS marketplace
- **Standard compliance without compromises**
  - ✓ From OWL and SHACL to RDF-star and graph-path search in SPARQL
- **Regular release cycle, stable quality & clear roadmap**
  - ✓ Robust dev. process allows for 5+ releases per year, rear bugs and no performance degradation

ontotext

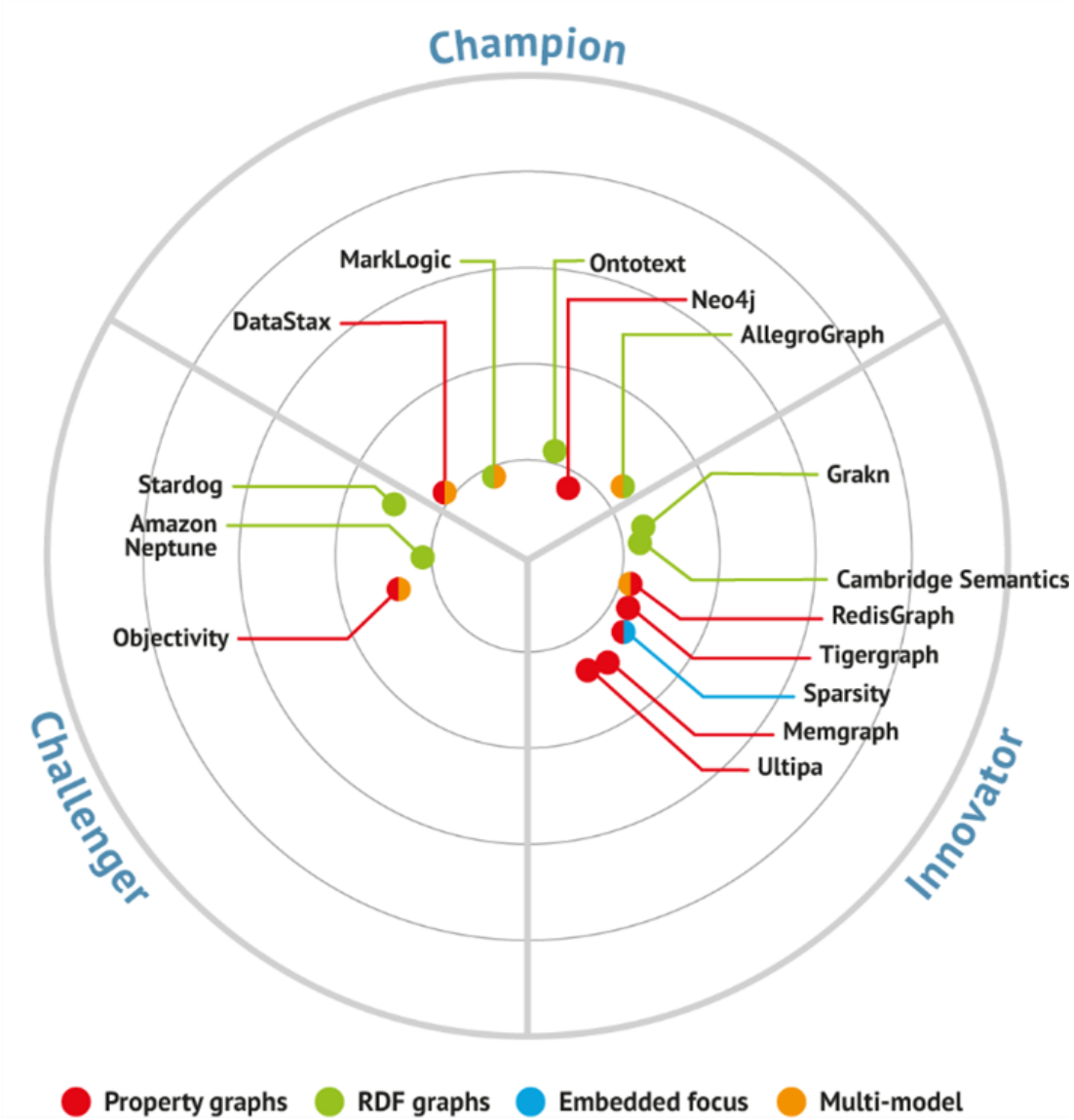
GraphDB™: Potent Semantics and Rich Ecosystem

- **Scalable reasoning** across the full lifecycle of the data
  - ✓ **Efficient reasoning**: the only engine offering efficient inference across all CRUD operations
    - Stardog's backward-chaining fails when querying big data; ORACLE requires recompute upon update
  - ✓ **SHACL data validation** which does work well on sizeable transactions
- **Variety of analytics & search capabilities**
  - ✓ **Integration with ChatGPT:** SPARQL functions to query LLMs, Query and result explanation
  - ✓ **Text analysis**: built-in pipelines for entity linking against large knowledge graphs
  - ✓ **Semantic similarity search** based on word- and graph-embedding
- **The richest set of seamlessly integrated partner tools**
  - ✓ **Knowledge management**, taxonomy & vocabulary management, ontology editors
  - ✓ **Data management**: data catalogues, ETL, data linking, graph & instance data editors
  - ✓ **Search**, exploration, visualization, chatbots

KGs + LLMs = Innovation

- **GraphDB integrates with ChatGPT**
  - Enhances user interaction with RDF, catering to both novices and experts
  - Simplifies diving into RDF, leveraging ChatGPT to uncover deeper insights from the data
- **Query and result explanations**
  - One can get textual summary of the results, instead of just a table of strings and numbers
  - Brings clarity on the underlying logic
  - Makes RDF and SPARQL easy to understand
- **Ask generic questions to ChatGPT from SPARQL**
  - Enrich your graphs and extract additional information from your data
  - Well thought out set of SPARQL functions allow for efficient processing of the data in the graph using LLM

GraphDB™ - Popular & Highly Prized by Analysts



100,000+ unique GraphDB installations LTM

50,000+ new installations in the last 90 days

*… the **market leaders in this space continue to be Neo4J and OntoText (GraphDB)**, which are graph and RDF database providers respectively. …*

Source: Graph Database Market Update 2023, Bloor Research
ttps://www.bloorresearch.com/technology/graph-databases/

Ontotext is referenced in 20+ Gartner reports in 2022-2023

Market Guide for Graph DBMS;
Hype Cycles for: DM; AI; Finance Data and Analytics Governance; NL Technologies, ….

ontotext

- Intro & Ecosystem
- Essentials
- High Availability & Security
- Workbench & Visualizations
- Search & Analytics
- Reasoning & Validation
- Edge Properties, RDF-star
- Graph Path Search
- Geographic Query Extensions
- Benchmarking
- Latest Releases

GraphDB Essentials

- **Scalable and Dependable RDF 1.1 engine**
  - Predictable performance across wide range of workloads
  - ACID compliant transactions
- **Platform Independent (Java)**
- **W3C Standards Compliant**
  - Comprehensive support for RDF, SPARQL, OWL 2, and SHACL
- **Reasoning and Validation**
- **High-Availability Cluster and Enterprise-Grade Security**
- **Connectors for Upstream and Downstream Integration**
- **Excellent Technical Support**

GraphDB Editions

FREE

Delivers a fully functional database optimized for desktop use and small commercial prototypes:

- No constraints on data scale
- Limited to one concurrent read
- Single inference thread
- Lucene connector for indexing

ENTERPRISE

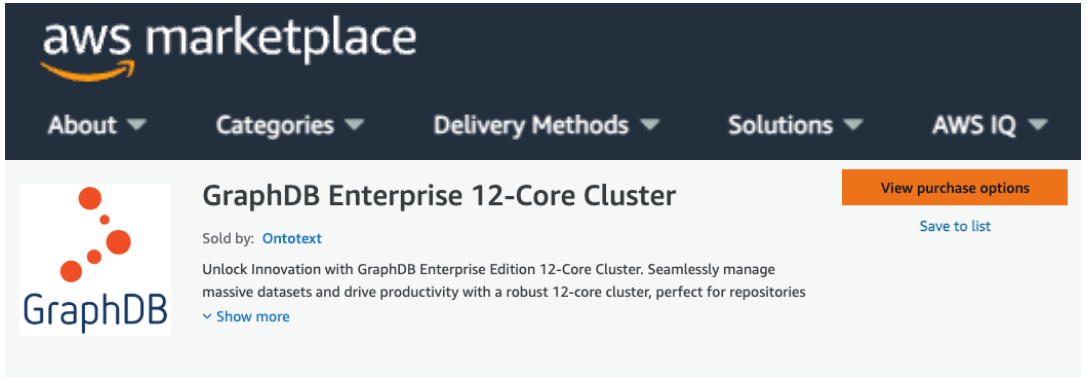
Offers cluster support for enterprise resilience and high-availability:

- No single point of failure
- Multi-data center support
- Unlimited scalability of the read operations
- Elasticsearch, OpenSearch, Solr and Kafka connectors

Multiple Operation and Deployment Options

- **Operation mode**
  - Self-service
    - On-prem/Cloud
    - Technical support with SLA
  - Managed service
    - AWS / Azure (coming soon) marketplaces
    - Service availability with SLA
- **Deployment options**
  - Native installation packages (zip, msi, dmg, deb, rpm)
  - Docker/Helm charts
  - AMI
  - Azure VM (coming soon)

	GDB Self-hosted	GDB Self-hosted via Marketplace	GDB Managed Service
Channel	Ontotext Direct	AWS / Azure	AWS / Azure
SLA/Dev support	Recommended	Recommended	Required
Hosting	Client	Client	Ontotext
Operation	Client	Client	Ontotext

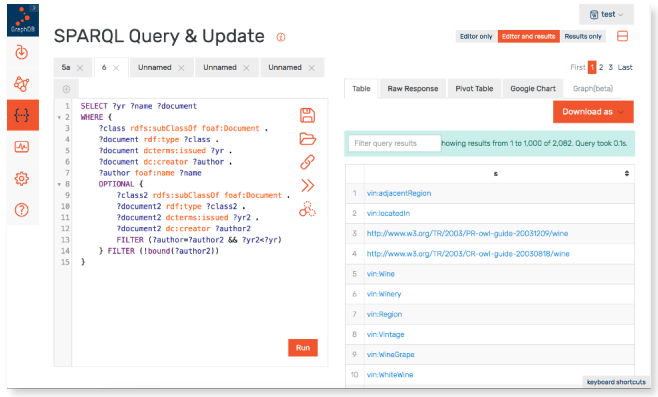


ontotext

Architecture

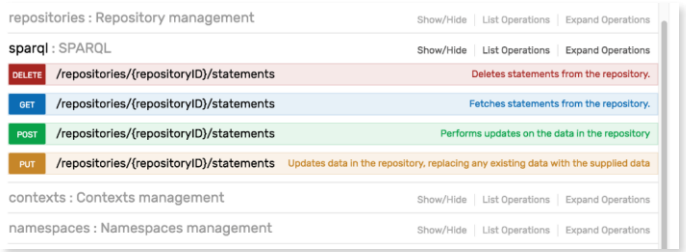
GraphDB Workbench

- User friendly interface for database administration



GraphDB Engine

- Storage
- REST API for database access



Plugins

- Various extensions
- Connectors

ontotext

Downstream Integration

- **JDBC Driver**
  - Provides SQL access to GraphDB, can be used for data analytics via Tableau and Power BI
- **Connectors for faceting and full-text search**
  - Elasticsearch
  - OpenSearch
  - Solr
  - Lucene

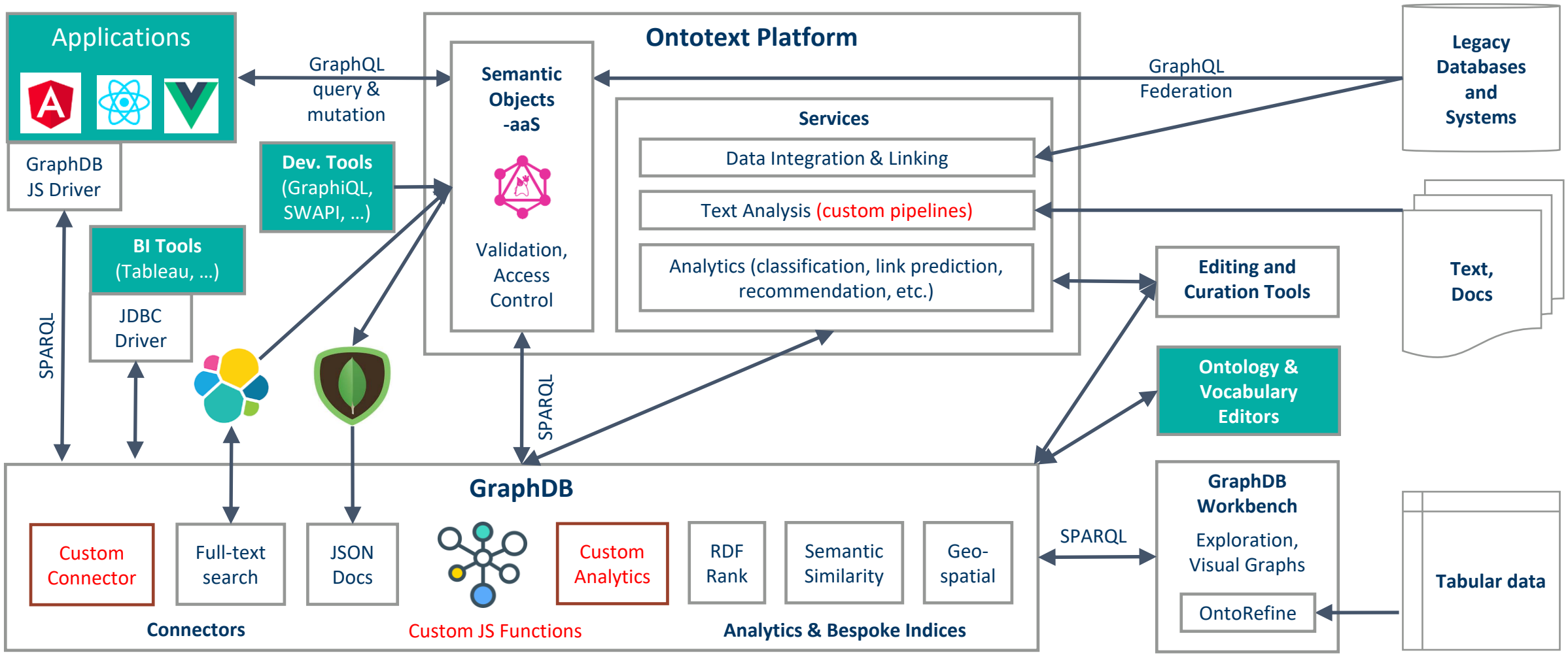
- **Connectors for general interfacing with other systems**
  - Kafka

Upstream Integration

- **Ontop integration**
  - Convert an SQL database to virtual RDF and SPARQL on the fly
- **MongoDB Connector**
  - Access to data stored in MongoDB via SPARQL

- **Text Mining plugin**
  - Extract knowledge from your textual data with annotations via a third-party service
- **Kafka Sink Connector**
  - Stream data from any system via Kafka to GraphDB
  - Use Smart Updates and SPARQL templates

Access, Customization and Integration Points

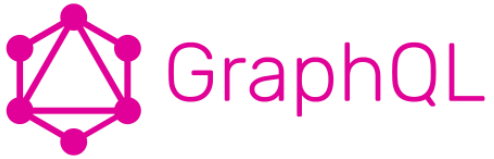


Ingest and Update

- **RDF 1.1** with RDF-star extensions
- **Reasoning** (inference) derives new knowledge
- **SHACL validation** ensures data quality
- **Ingest data** from other systems
  - **Ontop**: virtual RDF from SQL sources
  - **OntoRefine**: data from generic table-based sources
- **Automate complex updates** with the Kafka Sink Connector

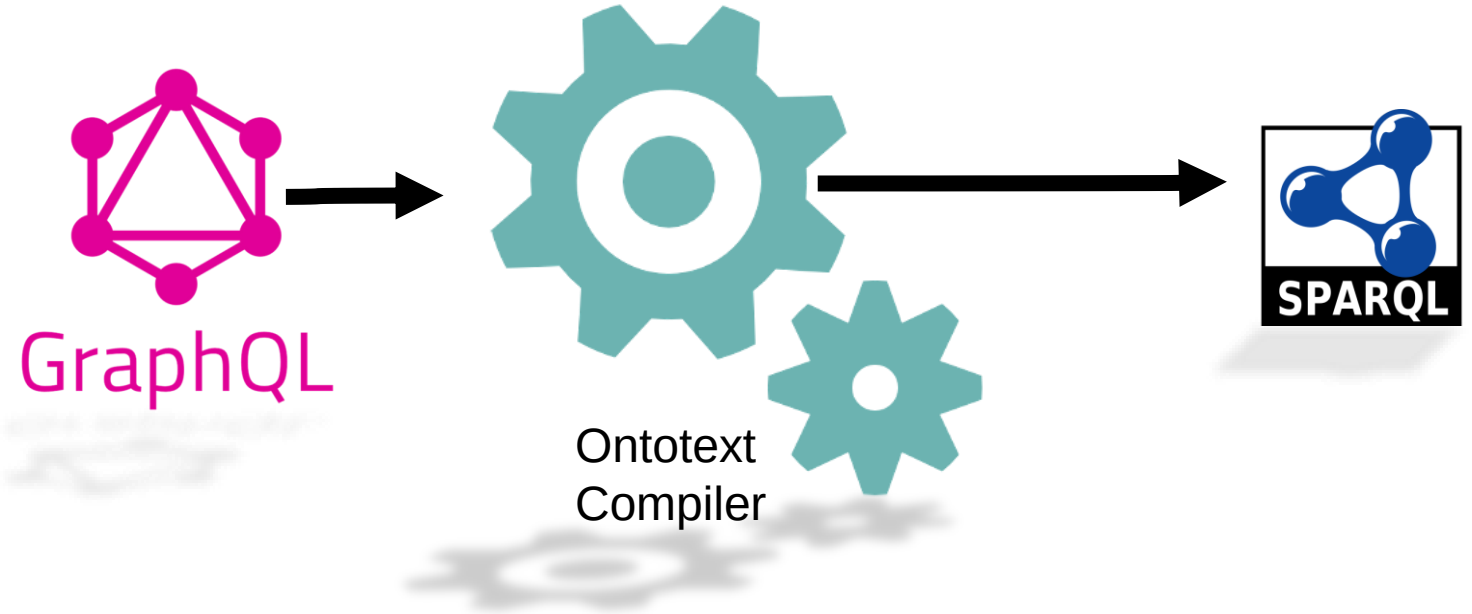
Query, Search and Retrieve

- **SPARQL 1.1** with SPARQL-star extensions
- **Many SPARQL extensions** that provide additional functionality
- **GraphQL**
- **Beyond vanilla SPARQL**
  - **Faceting and full-text search**
  - **Graph path search**
  - **Geographic query extensions**
  - **Ranking of RDF nodes**
  - **Similarity search**



GraphQL Access via Semantic Objects

- **Knowledge Graph access and updates via GraphQL**
- **Data validation via RDF Shapes**
- **Semantic Business Objects definitions done by business analysts**
  - GraphQL Schema and shapes generated from Semantic Objects



Develop with GraphDB

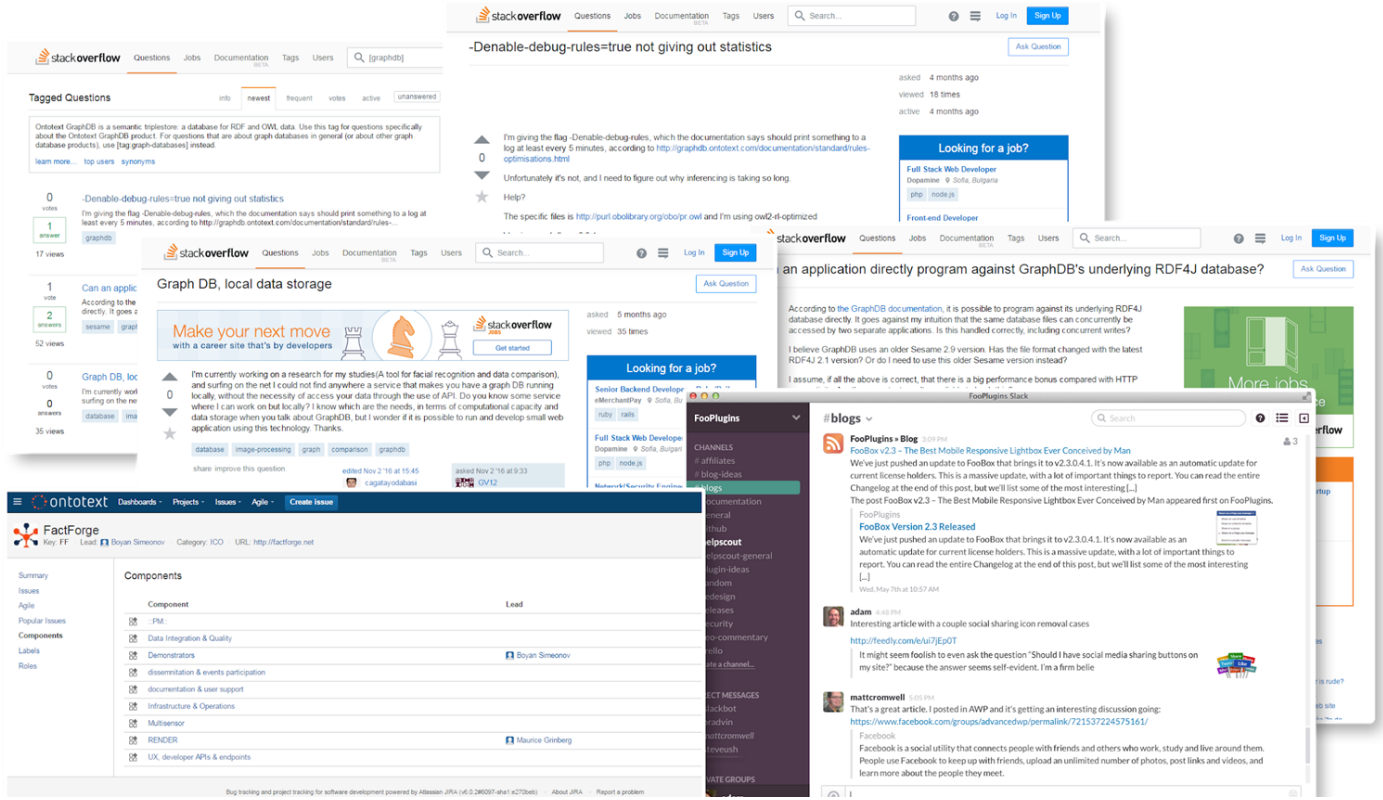
- **REST APIs**
- **RDF4J API and Java Client**
  - Ontotext is a major contributor to RDF4J
- **JavaScript Client optimized for node.js**
- **SPARQL Sequences** plugin
- **Custom JavaScript Functions in SPARQL**
- **Powerful Plugin API**
  - Provides low-level access to internal structures
  - Can be used to extend GraphDB

Deploy Anywhere

- Platform Independent (Java)
- Desktop Installation Packages
- Docker
- Kubernetes
- Cloud agnostic – AWS, Azure, Google Cloud, etc.
- Cloud on-demand offerings and managed services (AWS & Azure)

GraphDB Support

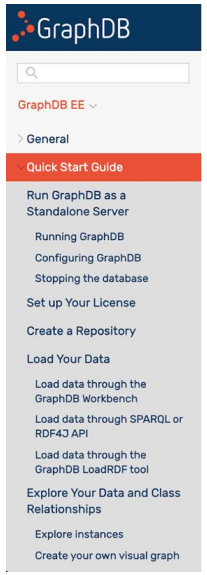
- **Dedicated Support Team**
- **Community Support**
  - Stack Overflow monitoring
- **24x7 Service Desk**
  - Jira Issue-Tracking System
- **Fully Managed Service and Custom SLAs (optional)**
- **Customized Runbooks**
- **Easy Slack Communication**

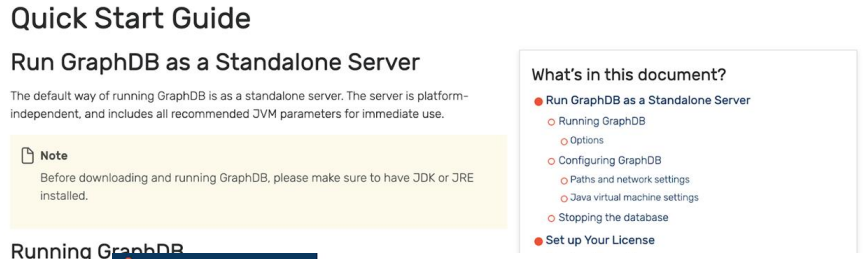


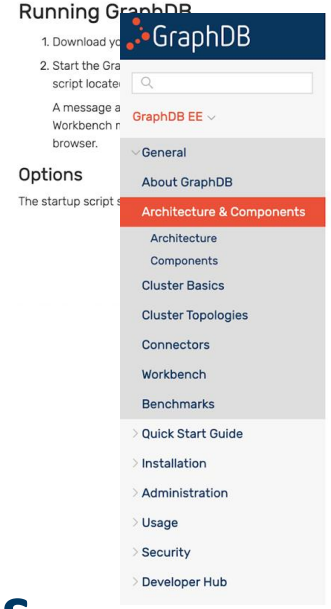
ontotext

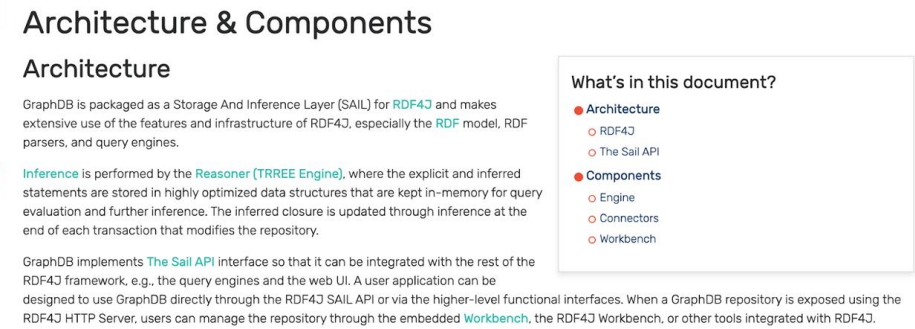
GraphDB Documentation & Training

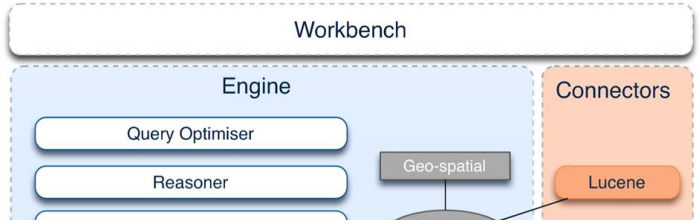
- Always up-to-date
- Dedicated Doc team
- Startup Guide
- Interactive user guides
- How-To Examples
- Video Tutorials
- Webinars and Live Trainings
- Online and on-site training courses









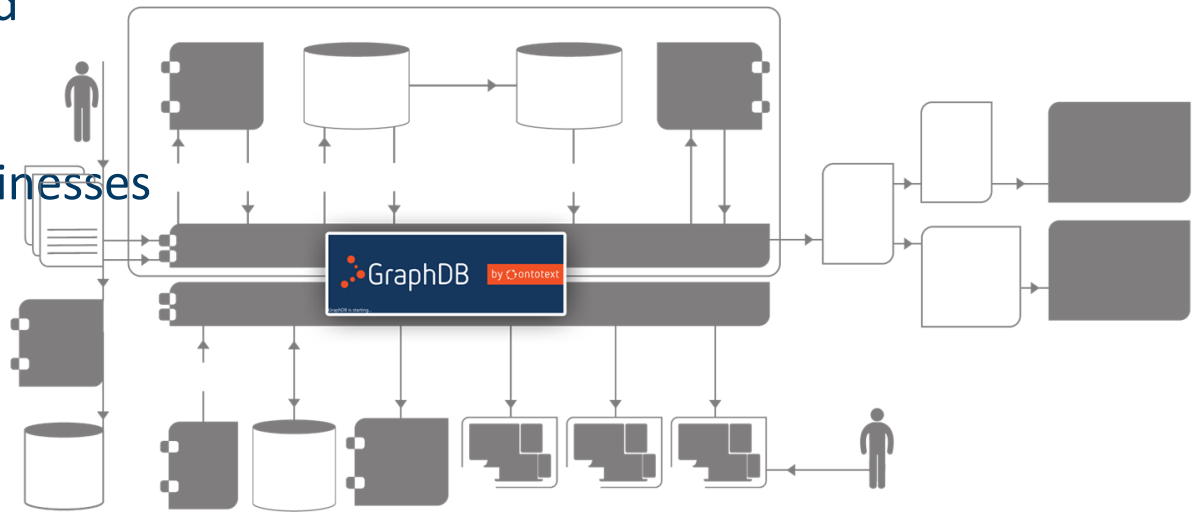


ontotext

- Intro & Ecosystem
- Essentials
- High Availability & Security
- Workbench & Visualizations
- Search & Analytics
- Reasoning & Validation
- Edge Properties, RDF-star
- Graph Path Search
- Geographic Query Extensions
- Benchmarking
- Latest Releases

GraphDB – Stability for Mission-Critical Apps

- **High-Availability cluster constantly improving over 12+ years**
- **Backup and Restore with S3 support**
- **Monitoring API compatible with Prometheus**
- **Powering critical systems across multiple industries**
  - Fin. Services: Top-5 banks in USA and Swizerland
  - AECO: Johnson Controls, Schneider Electric
  - Manufacturing: Siemens, 3 of the Top-5 car businesses
  - Life Sciences: Roche, AstraZeneca, SBI
  - Market Intelligence: S&P
  - Government: UK Parliament, NASA
  - News and Media: FT, BBC



ontotext

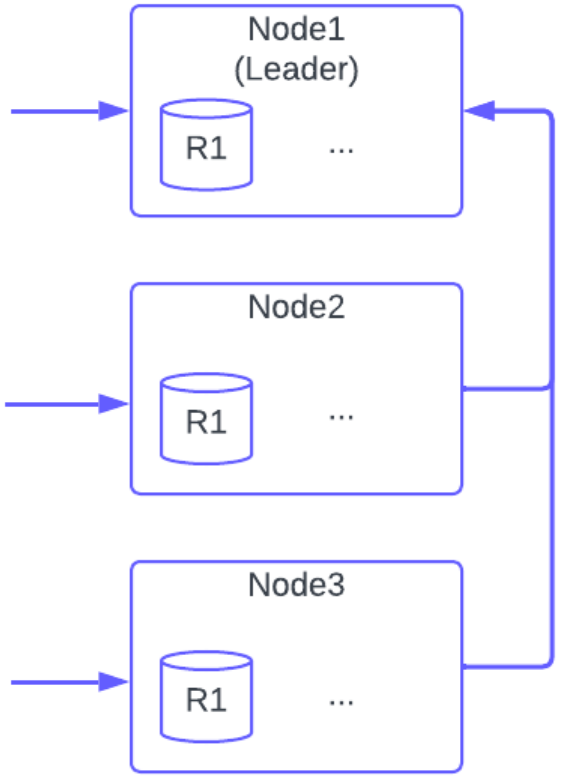
High Availability Cluster Architecture

- Based on the Raft consensus algorithm
- Coordinates all read and write operations
- Ensures that all worker nodes are synchronized
- Propagates updates across all nodes and checks updates for inconsistencies
- Load balances read requests between all available nodes

ontotext

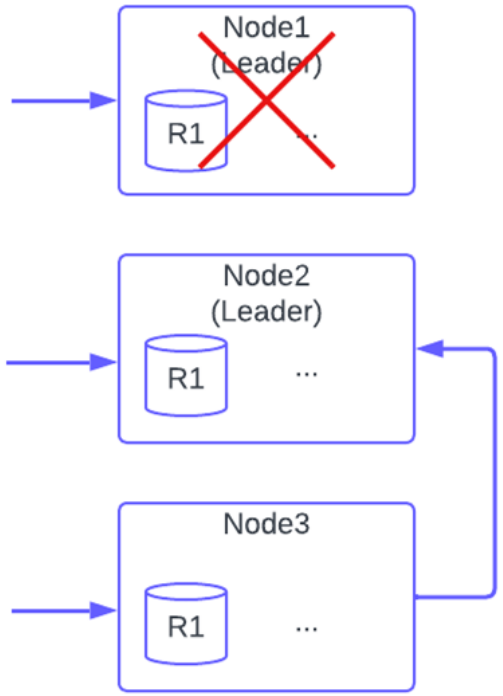
High Availability Cluster Architecture (ctd)

- **Improved resilience**
  - Failover, dynamic configuration
- **Improved query bandwidth**
  - Larger cluster means more queries per unit time
- **Deployable across multiple data centers**
- **Elastic scaling in cloud environments**



High Availability Cluster Architecture (ctd)

- Automatic leader election to support recovery from any failure
- A smart client supporting multiple endpoints
- External proxy provides the smart client layer to any RDF4J-compatible client



Enterprise-Grade Security and Access Control

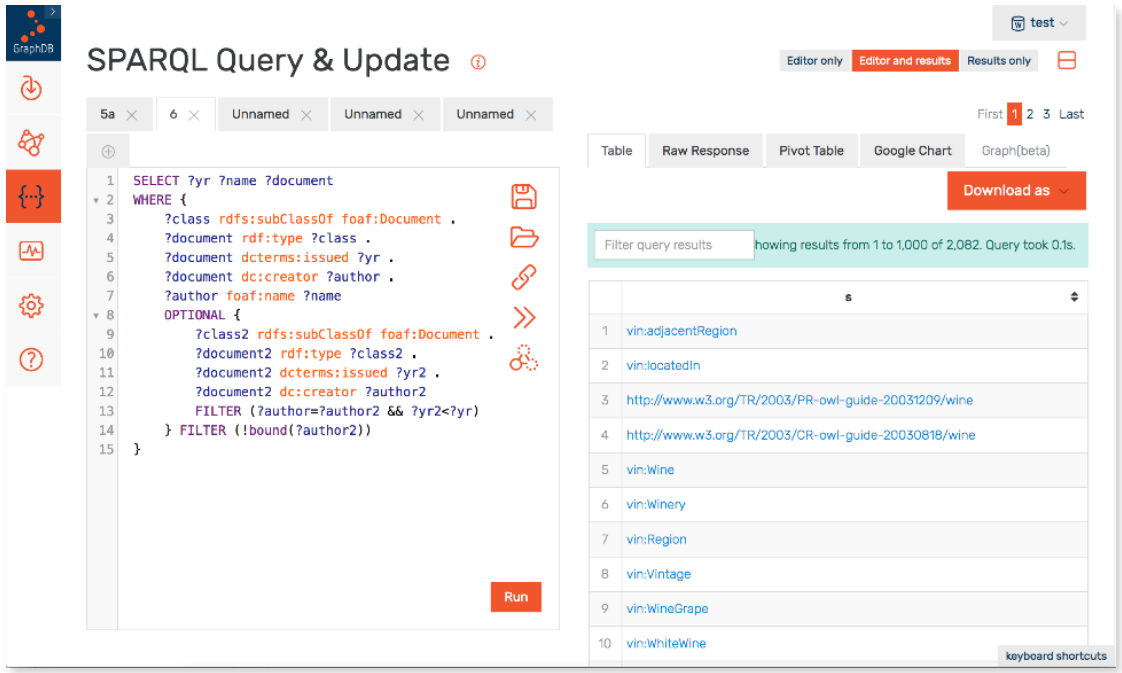
- OpenID/OAuth
- LDAP
- Kerberos
- X.509 Authentication
- Role-Based Access Control (RBAC)
- Encryption in transit

ontotext

- Intro & Ecosystem
- Essentials
- High Availability & Security
- Workbench & Visualizations
- Search & Analytics
- Reasoning & Validation
- Edge Properties, RDF-star
- Graph Path Search
- Geographic Query Extensions
- Benchmarking
- Latest Releases

Explore and Manage with GraphDB Workbench

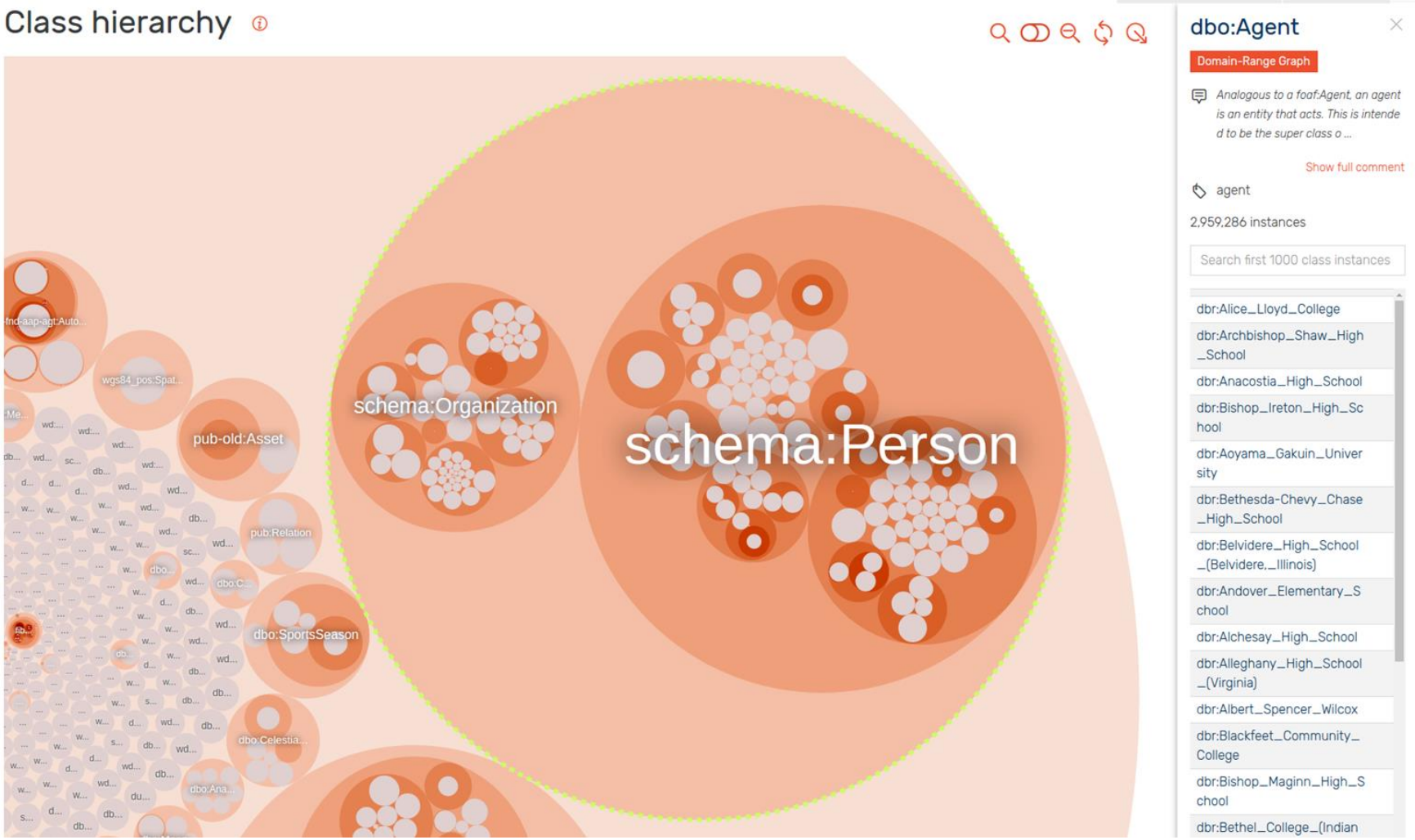
- **Run SPARQL**
- **Import Data**
- **Explore and Visualize**
  - Autocomplete
  - Class hierarchy
  - Class relationships
  - Visual graph
- **Monitor and Administer GraphDB**
- **Run Interactive Guides**



ontotext

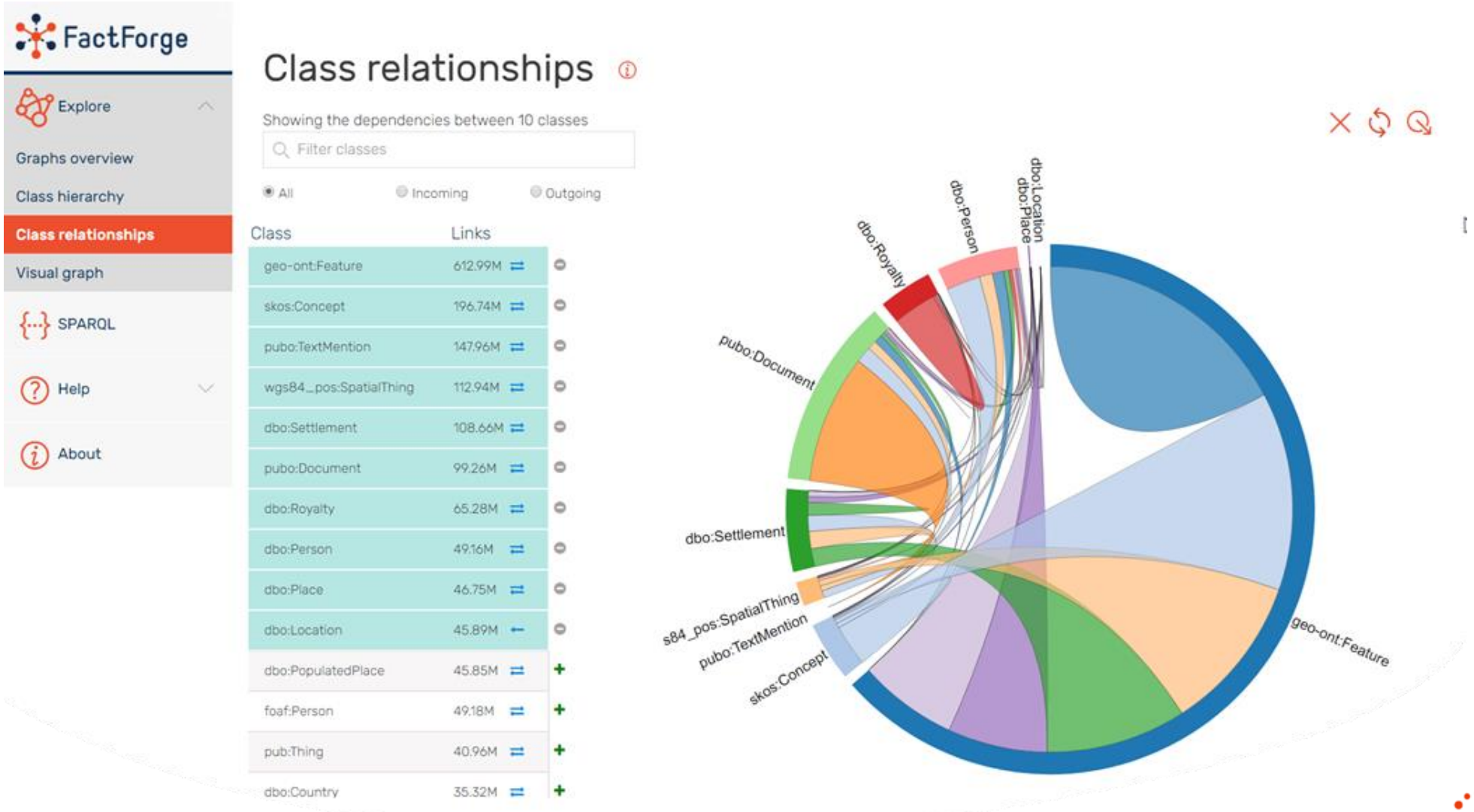
Class Hierarchy Exploration

- **Explore ontologies of 1000+ classes**
- **Get sample instances**



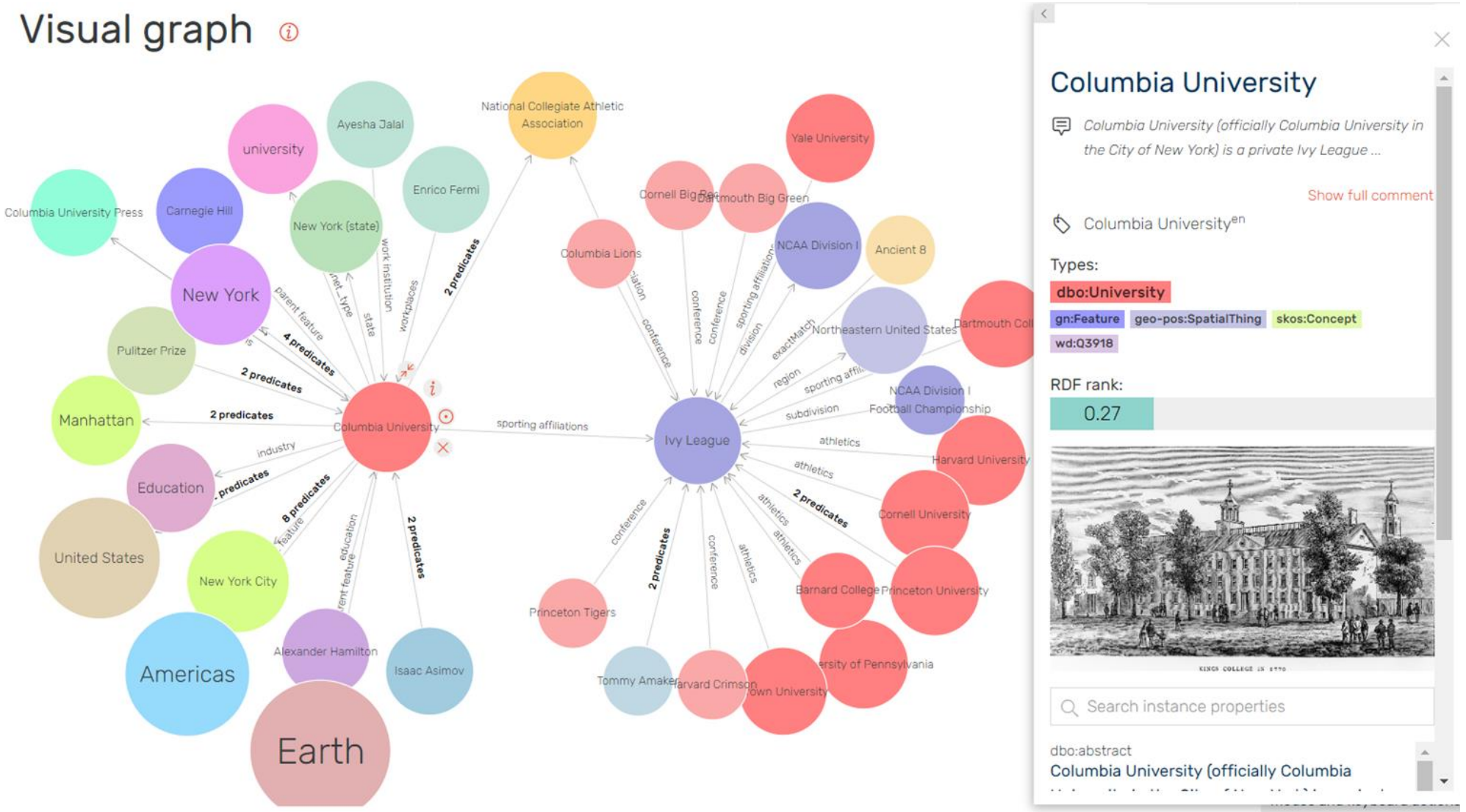
ontotext

Explore Relationships Between Classes



ontotext

Explore Nodes with Visual Graph



ontotext

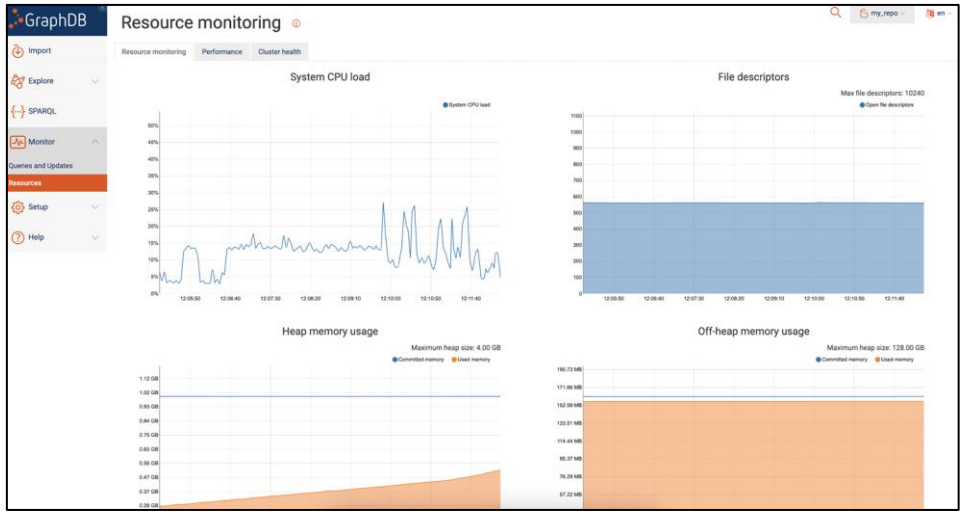
Explore Nodes with Visual Graph

- **Customize the Visualization**
  - Filter out abstract resources
  - Include/exclude specific properties
  - Generate new relationships on the fly
  - Show/hide inferred statements
- **Better views via node importance**
  - Use GraphDB-s RDFRank plugin to get the importance/centrality rank of a node
  - Use importance to chose the top-20 related nodes to be shown (configurable)
  - Use importance ranks to size the nodes

ontotext

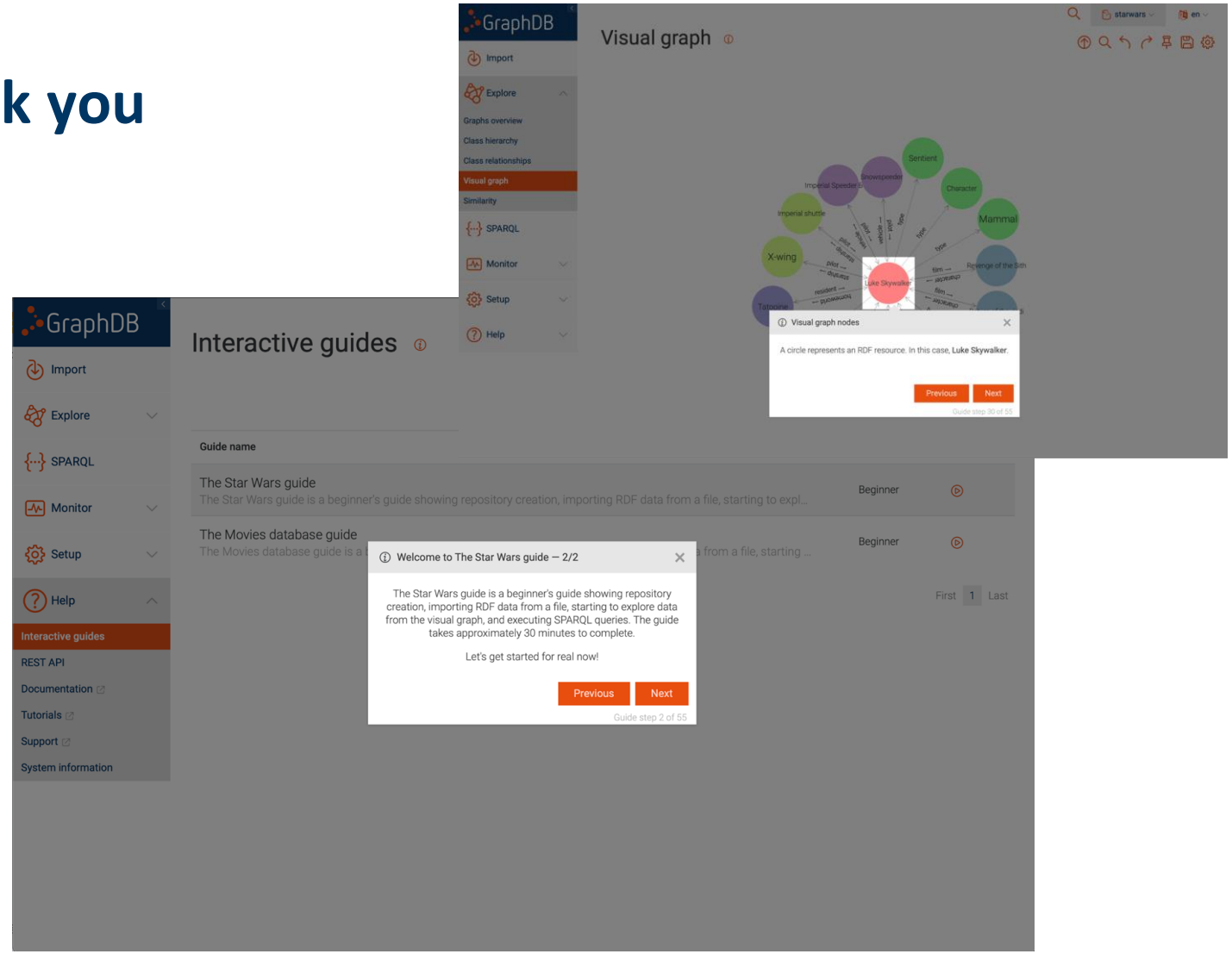
Monitor and Administer GraphDB

- **Monitoring**
  - System resources
  - Running operations
  - Cluster status and health
- **Administration**
  - Manage repositories
  - Manage users and roles
  - Create cluster
  - Create connectors



Run Interactive Guides

- **Interactive tutorials that walk you through key GraphDB functionalities using the Workbench user interface**
  - Creating repositories
  - Importing data
  - Exploring data with visualizations
  - Running SPARQL queries



ontotext

- Intro & Ecosystem
- Essentials
- High Availability & Security
- Workbench & Visualizations
- Search & Analytics
- Reasoning & Validation
- Edge Properties, RDF-star
- Graph Path Search
- Geographic Query Extensions
- Benchmarking
- Latest Releases

ontotext

Search and Analytics Features

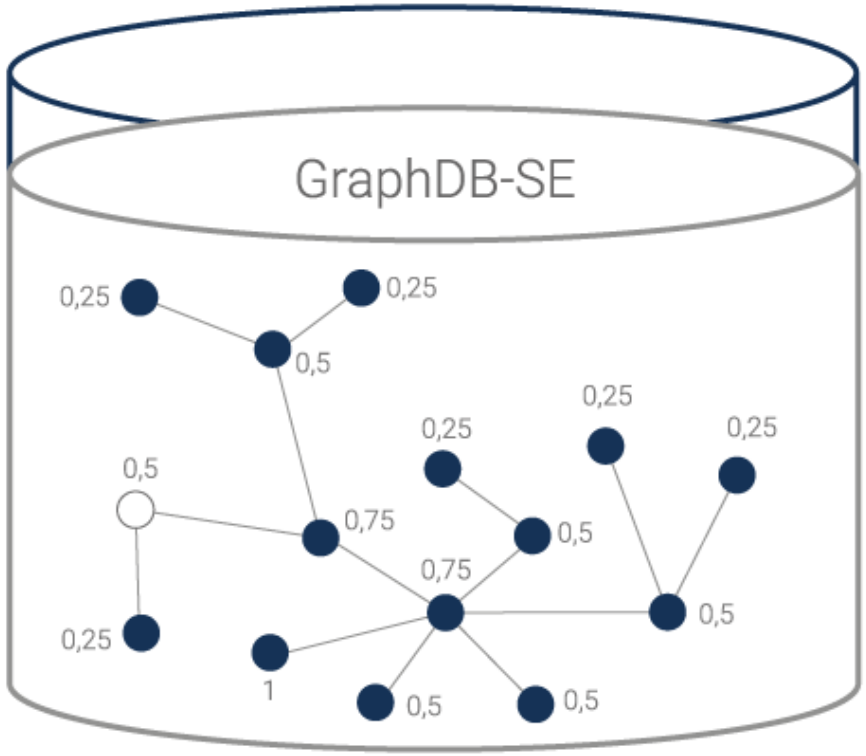
- **Faceting and full-text search**
  - **Faceting and FTS connectors:** search the KG via Lucene, Solr, Elasticsearch or OpenSearch
  - **Simple FTS:** basic full-text search at the literal level
- **RDFRank**: node importance/centrality calculated via PageRank
- **Text mining plug-in**
  - Allows for extraction of facts and metadata from unstructured content, e.g. documents
  - Integrated multiple external text analysis services, e.g. spaCy server, GATE Cloud, Ontotext's Tag API
- **Semantic Similarity** based on word and graph-embedding
- **Graph Path Search**
- **Geographic Query Extensions**

Full-Text Search Connectors

- **Full-Text Search (FTS):** Enables quick, precise and flexible text-based queries, uncovering relevant data
- **Faceting:** Classifies results into categories, offering an organized view and facilitating efficient data filtering
- **Seamless integration with popular full-text search engines**: Lucene, Solr, Elasticsearch, OpenSearch
- **Precise selection and filtering** of what data will be indexed. Multiple FTS indices can be defined for one and the same graph
- **Ranking of results can be boosted graph analytics** (see RDFRank)
- **Changes in the graph are reflected automatically**, indices are always up-to-date

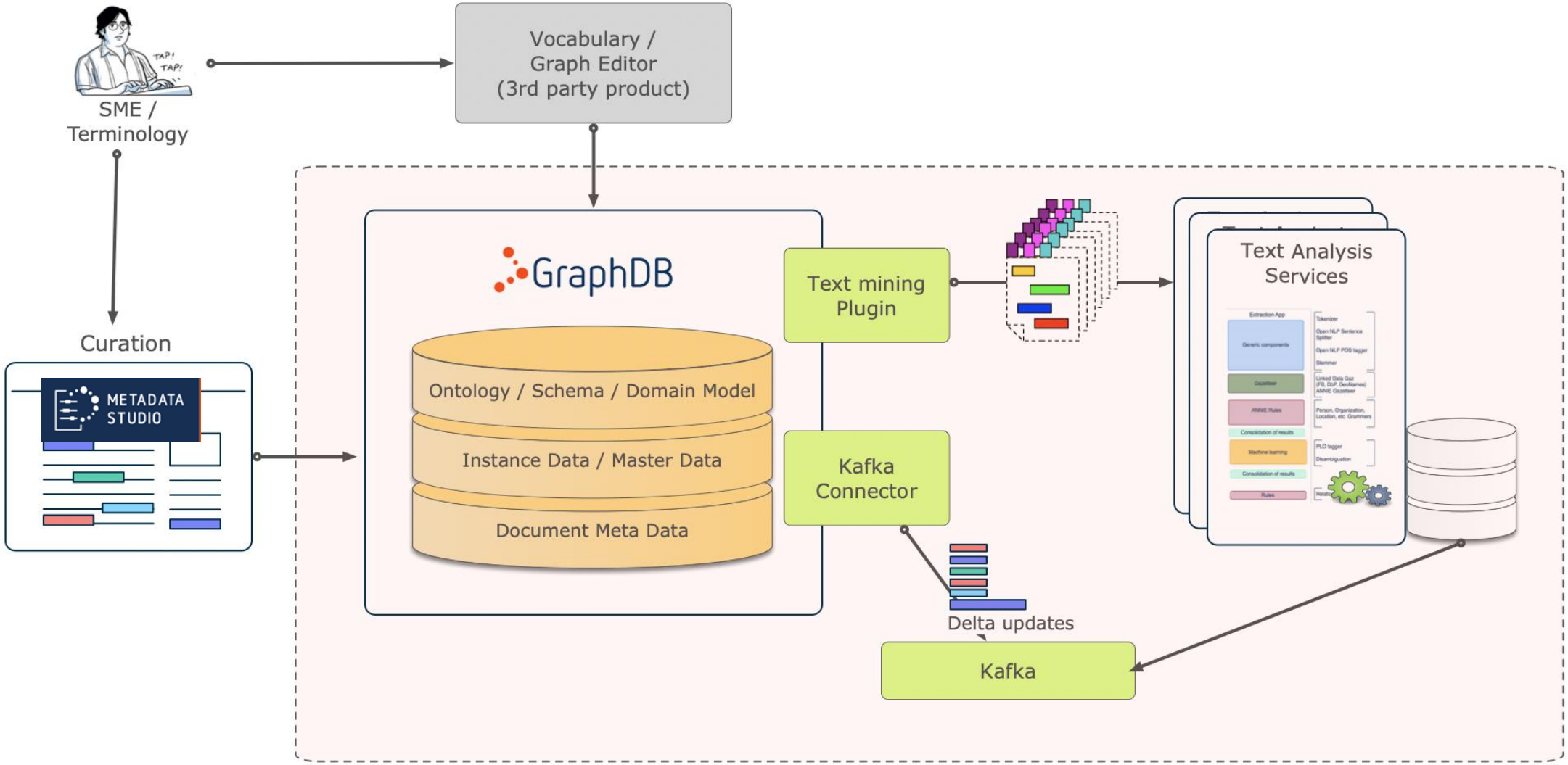
RDFRank: The Importance of a Node in a Graph

- **GraphDB's RDFRank computes PageRank**
  - Calculates "importance" based on nodes' interconnectedness
  - Node ranks accessible via the rank:hasRDFRank predicate
  - Incremental RDF Rank calculation upon update is useful for dynamic data
- **RDF Rank is used in GraphDB Workbench**
  - For ordering big lists of auto-suggest options in the SPARQL Editor and Search
  - To determine the node size of GraphDB's Visual Graph
  - As feature for similarity ranking



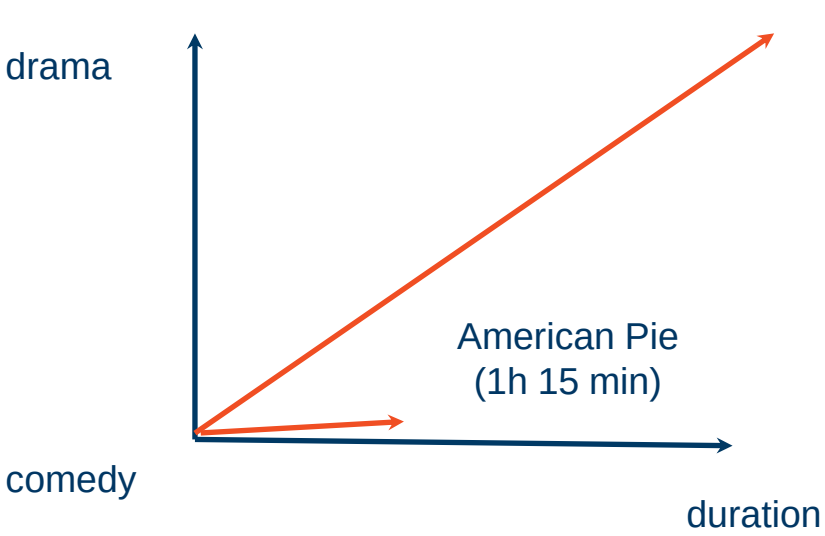
RDF Ranking Computation

Text Analysis Architecture Powered by GraphDB



Knowledge Graph Embedding?

- Predict similar graph nodes or properties
- Require no input training data
- Mathematical representation of graph nodes as vectors



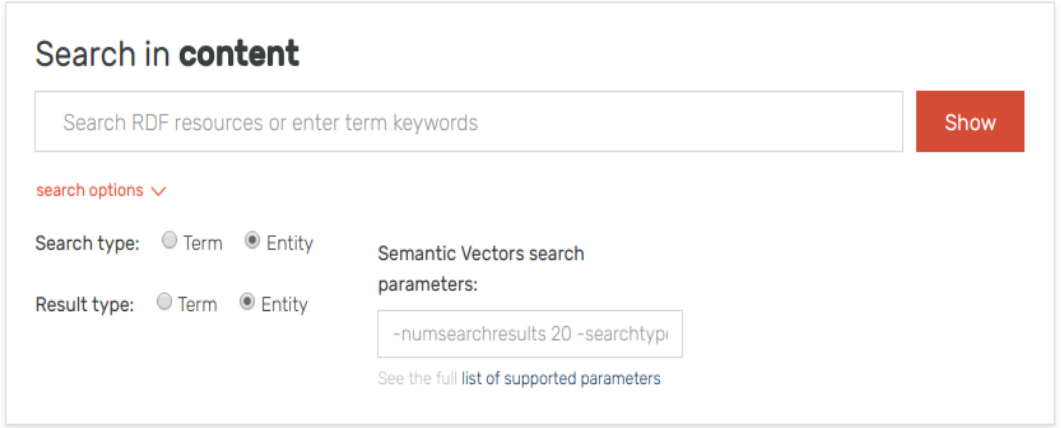
vs.

GraphDB Semantic Similarity Plugin

- **Statistics similarity on KG using Semantic Vectors**
  - Creates statistical semantic models from your RDF data
  - Uses Lucene for scalable indexing and search for similar terms and documents
  - Reduced dimension vector space, e.g., 200 or 2000
- **Can judge similar nodes based on similar edges**
  - Not only an exact match of predicate-object pairs of the basic VSM
  - Example: <locatedIn, Manhattan> and <hasOfficeIn, New York City>
- **Both text-based and graph/predication-based embeddings**
  - Most interesting: combining text embeddings and graph embeddings models

Similar News

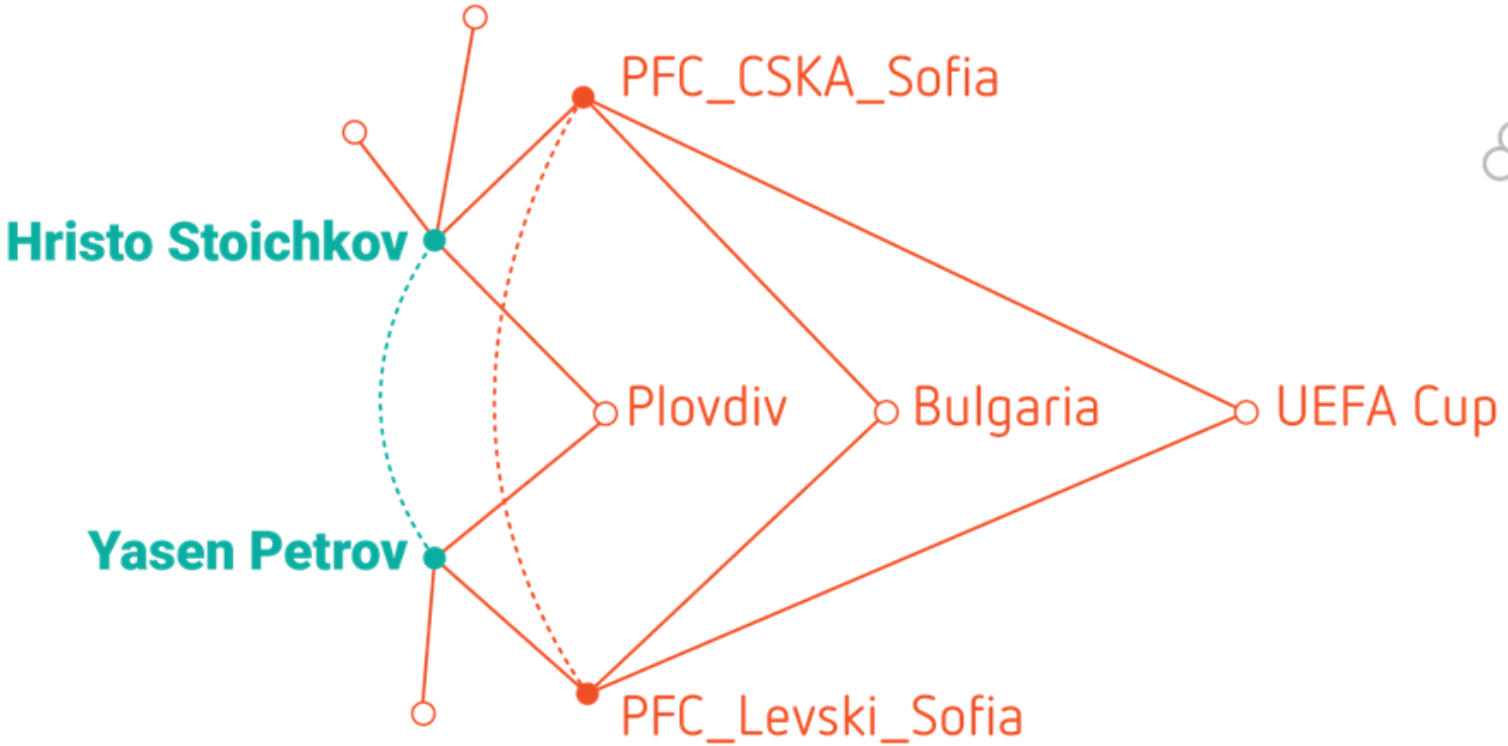
- **Task**
  - Given a news article
  - Get articles from other news sources that report the same event
- **Solution**
  - Create an index over the news in FactForge
  - Get: similar news, terms describing a news article, similar terms, …
- **Easy out of the box**
  - It takes 2h to index 1M news
  - No parameter tuning needed

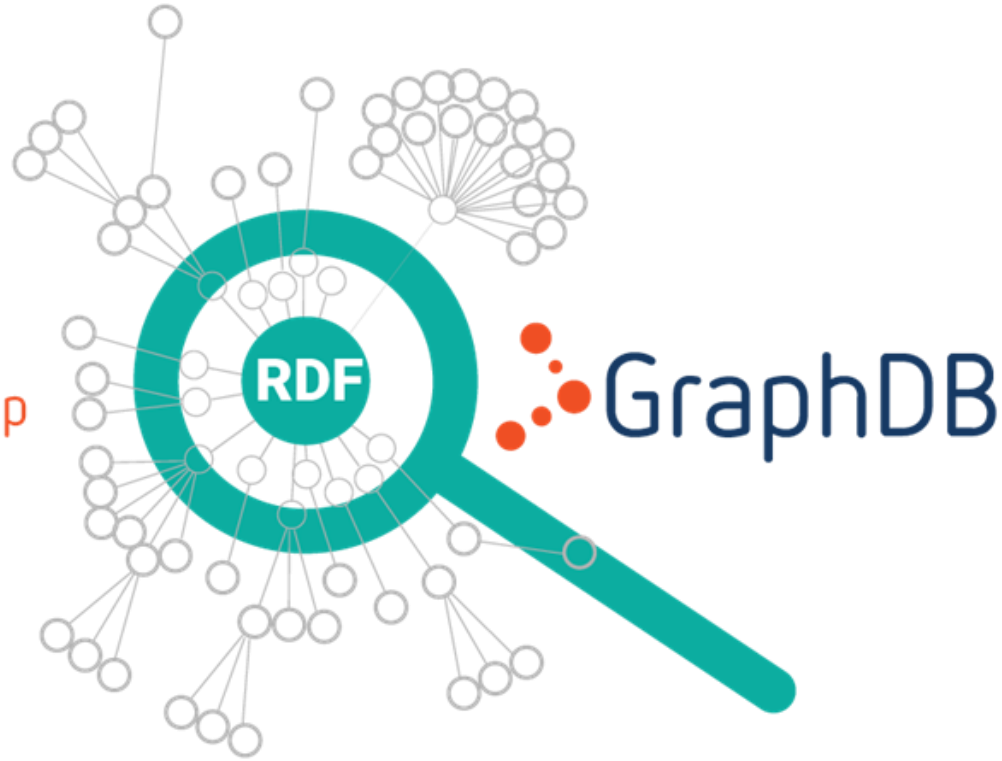


Showing results for https://www.cnbc.com/2018/07/17/us-senate-republican-leader-warns-russia-not-to-meddle-in-2018-electio.html View SPARQL Query

	entity	score
1	https://www.cnbc.com/2018/07/17/us-senate-republican-leader-warns-russia-not-to-meddle-in-2018-electio.html	"1.0"^^xsd:double
2	https://www.reuters.com/article/us-usa-russia-ryan-mcconnell/top-senate-republican-warns-russia-on-election-meddling-idUSKBN1KE2N5	"0.9297294882366121"^^xsd:double
3	https://www.politico.com/story/2018/08/06/rand-paul-russia-meeting-764589	"0.9083294545242895"^^xsd:double
4	https://www.sfgate.com/news/article/A-letter-from-Trump-to-Putin-is-the-latest-flash-13141621.php	"0.9032596414329951"^^xsd:double
5	https://www.cnbc.com/2018/07/18/trump-believes-next-meeting-with-putin-should-happen-after-the-russia.html	"0.9021681352123165"^^xsd:double
6	http://thehill.com/policy/defense/policy-strategy/399203-mattis-denies-policy-changes-made-at-trumps-meeting-with-putin	"0.8998150046751944"^^xsd:double
7	https://nypost.com/2018/07/18/majority-of-americans-disapprove-of-trumps-dealings-with-russia/	"0.8997648702930882"^^xsd:double

Similar Nodes Based on Graph Embedding





- Intro & Ecosystem
- Essentials
- High Availability & Security
- Workbench & Visualizations
- Search & Analytics
- Reasoning & Validation
- Edge Properties, RDF-star
- Graph Path Search
- Geographic Query Extensions
- Benchmarking
- Latest Releases

ontotext

Reasoning, Validation and Governance Features

- **Customizable RDFS and OWL reasoning**
- **SHACL Validation**
- **Inferencing tools**
  - **Proof** plugin: explain why something was inferred
  - **Provenance** plugin: run inference on a subset of the data
- **Track and understand changes**
  - **Change Tracking** plugin: track the changes that occurred within the transaction
  - **History** plugin: track the data changes across all transactions in time

Reasoning in GraphDB

- **Fast forward-chaining materialization**
  - Allows for efficient query evaluation on big datasets
- **Incremental for both inserts and deletes**
  - Inferred closure is updated transparently upon commit of transaction
- **Sample rules:**

ENTAILMENT

```
p  <rdf:type>  <owl:FunctionalProperty>
x  p  y
x  p  z
-------------------------------
y  <owl:sameAs>  z
```

CONSISTENCY

```
x owl:sameAs y
x owl:differentFrom y
------------------------
```

OWL 2 Reasoning

- **Built-in rulesets for: RDFS, OWL-Horst, OWL2-RL, OWL2-QL**
- **Optimized handling of owl:sameAs identifier mappings**
- **Custom rulesets easily defined**
  - Ruleset optimizer/profiler
- **Configurations with multiple rulesets**
  - E.g., one with consistency checking to be used for internal data and another one with „open-world“ semantics for LOD and other external datasets
- **Proof plugin provides inference explanation**

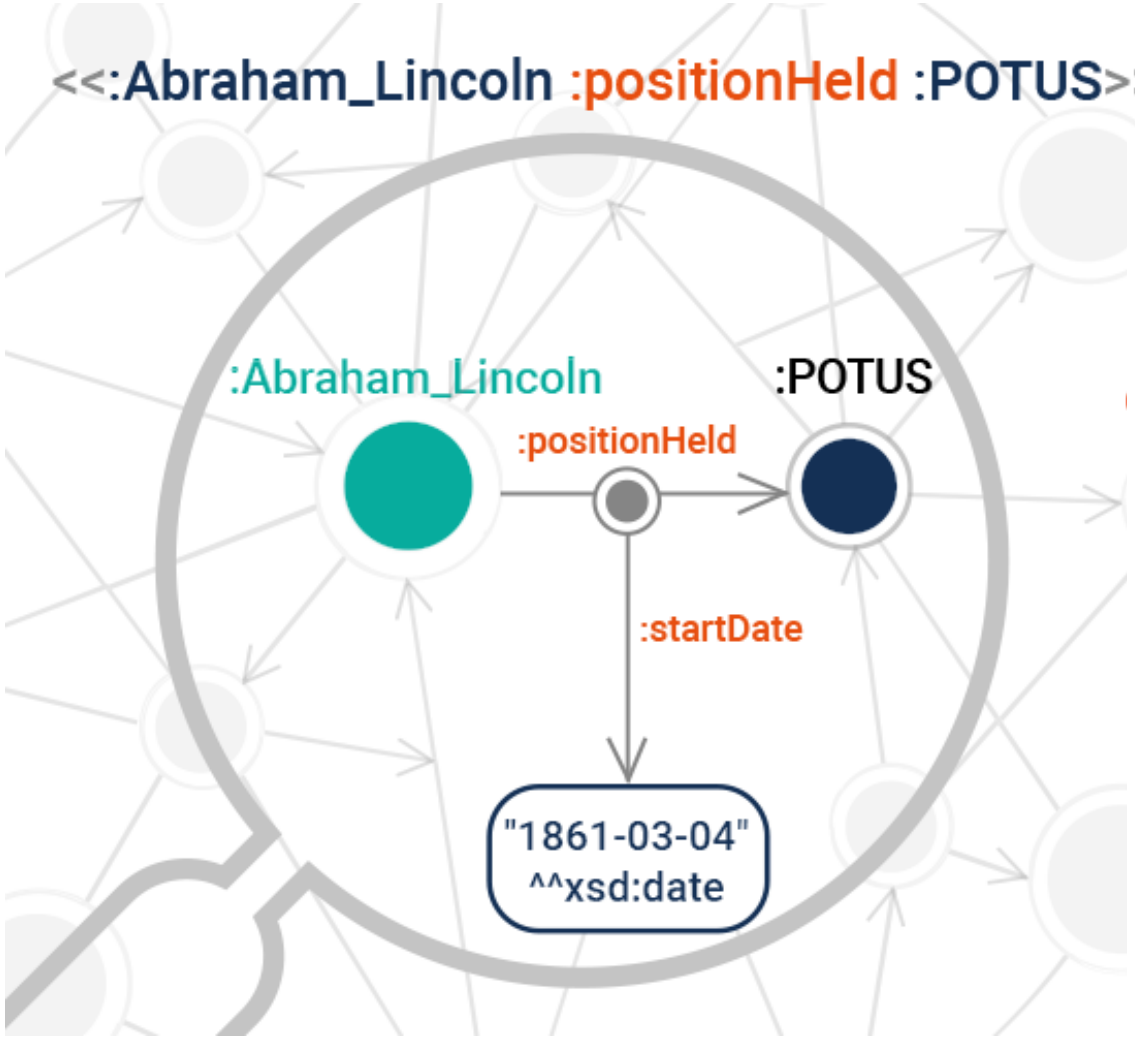
Efficient SHACL Validation

- SHACL is a W3C Standard for validating RDF
- Ensures data consistency and integrity
- Automates data validation
- Emphasizes validation over reasoning unlike OWL 2

- Intro & Ecosystem
- Essentials
- High Availability & Security
- Workbench & Visualizations
- Search & Analytics
- Reasoning & Validation
- Edge Properties, RDF-star
- Graph Path Search
- Geographic Query Extensions
- Benchmarking
- Latest Releases

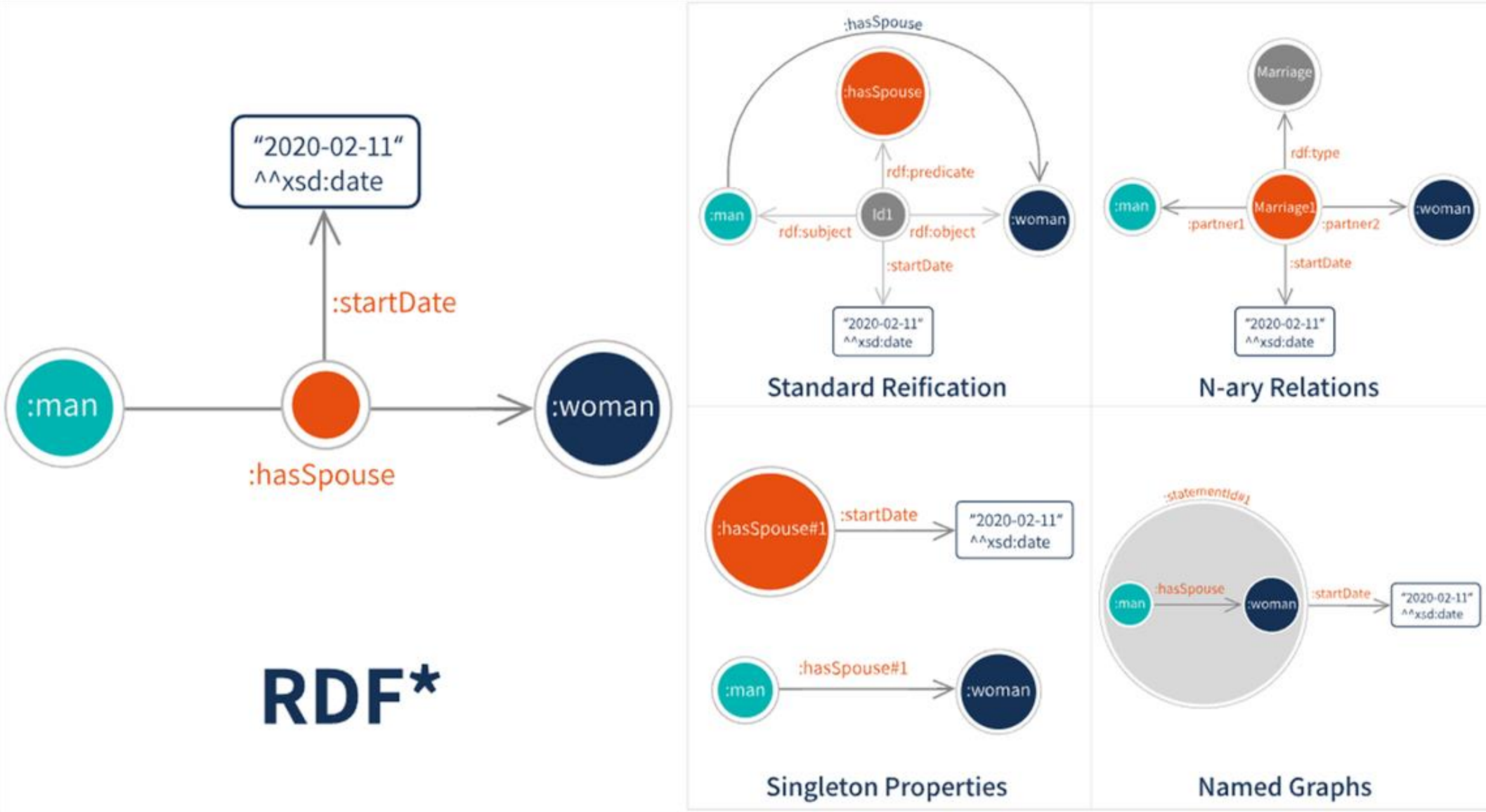
ontotext

RDF-star allows edge descriptions



- **Statements about statements**
  - ✓ Allows multiple level of nesting
  - ✓ Backward compatible
- **Much more expressive than the properties (key-value pairs) in LPG**
- **RDF-star is included in the upcoming RDF 1.2 standard**

RDF-star makes edge metadata simpler



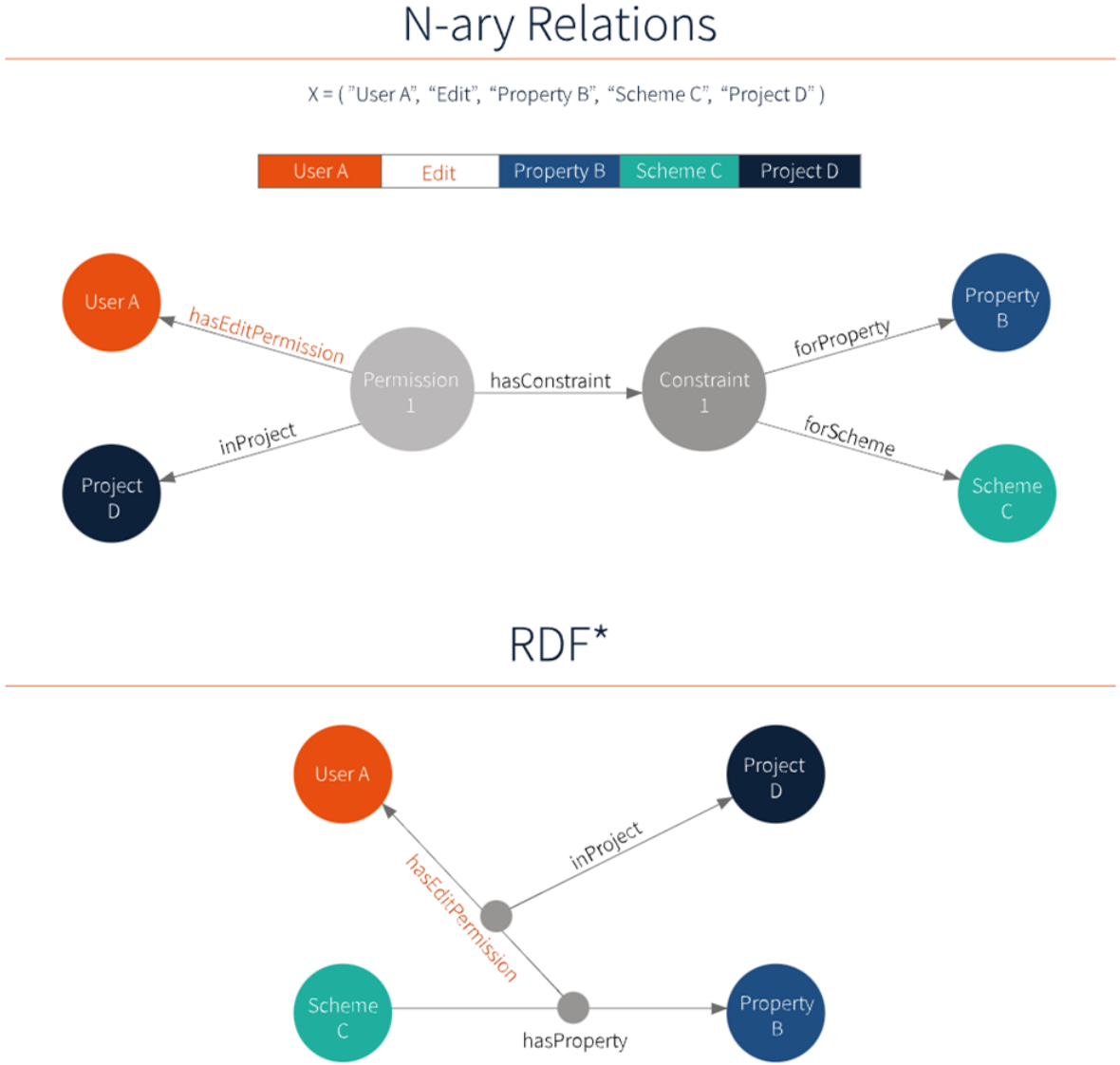
ntotext

More efficient metadata management with RDF-star

WikiData modeling approach	Statements count	Loading time (min)	Repository image size (MB)
Standard reification	391,652,270	52.4	36,768
N-ary relations	334,571,877	50.6	34,519
Named graphs	277,478,521	56	35,146
RDF-star	220,375,702	34	22,465

Note: Used a dataset testing all modelling approach with Wikidata from the authors of “Reifying RDF: What works well with wikidata?”

Use case: Access control for vocabulary management



- Synaptica is Ontotext partner
- Graphite is one of the leading vocabulary management tools
- Graphite uses GraphDB to manage SKOS data
- Synaptica re-implemented the Access Control Lists of Graphite using RDF-star and gained
  - ✓ Performance
  - ✓ Simplicity

ontotext

GraphDB Has The Best RDF-star Support

Feature	GraphDB 9.2	Stardog 7.1	AnzoGraph DB 2.0	Blazegraph 2.1
Nested embedded triples	**Yes**	**No**	**No**	**Yes**
Reference to a non-existing statement	**Yes**	**No**	**No**	**No**
Duplicate standard reification with embedded triples	**Yes**	**Yes**	**No**	**Yes**
Backward and forward compatibility	**Yes**	**No**	**No**	**Partial**
Other issues		Cannot handle embedded triple in object position	Could not use embedded triples in BIND clause	Cannot return statements as variables

Note: ORACLE and AWS Neptune does not support RDF-star

- Intro & Ecosystem
- Essentials
- High Availability & Security
- Workbench & Visualizations
- Search & Analytics
- Reasoning & Validation
- Edge Properties, RDF-star
- Graph Path Search
- Geographic Query Extensions
- Benchmarking
- Latest Releases

ontotext

Graph Path Search Outlook

- Identifies routes between two nodes in the graph
- Enables complex queries to uncover deep insights
- Efficiently navigates interconnected data, revealing hidden relationships
- Supports various traversal algorithms and criteria

Graphs and Path Traversal

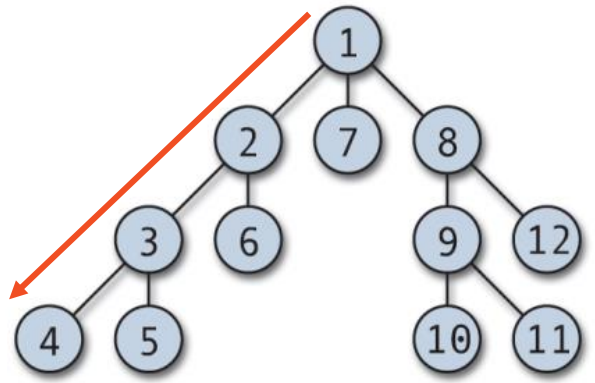
Common tasks:

- Check **if two nodes are connected**
- Find the **shortest path** between two nodes
- Find **all paths** between two nodes
- Find all **neighboring nodes** of distance X

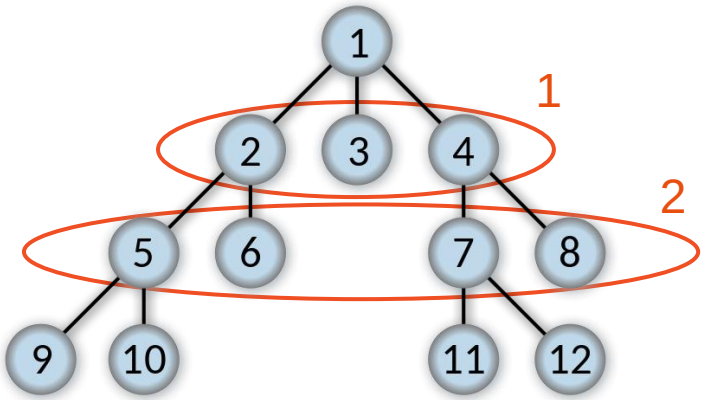
Well known computational challenge:

- BFS' time complexity is exponential WRT depth of search
- Requires lots of space/RAM

Depth First Search



Breadth First Search



ontotext

Graph path search use cases

- Road navigation
- Knowledge graph analysis
- Supply chain analysis
- Causality mining
- Recommendation
- Social network analysis

Limitations of vanila SPARQL for path search

- **Possible with SPARQL 1.1 property paths, HOWEVER:**
  - ✓ They uncover the start and end nodes, but not the intermediate ones
  - ✓ Shortest path is tough
- **Workarounds are ugly and slow**
- **To address this, all major triplestores made SPARQL extensions**

```
SELECT DISTINCT ?fr ?fr2 ?fr3 ?dist
WHERE {
    { SELECT DISTINCT ?fr (1 as ?dist)
      WHERE { sn:pers00000024189255811103 snvoc:knows/snvoc:hasPerson ?fr. }
    } UNION {
      SELECT DISTINCT ?fr ?fr2 (2 as ?dist)
      WHERE {
          sn:pers00000024189255811103 snvoc:knows/snvoc:hasPerson ?fr2.
          ?fr2 snvoc:knows/snvoc:hasPerson ?fr. }
    } UNION {
      SELECT DISTINCT ?fr ?fr2 ?fr3 (3 as ?dist)
      WHERE {
          sn:pers1 snvoc:knows/snvoc:hasPerson ?fr2.
          ?fr2 snvoc:knows/snvoc:hasPerson ?fr3.
          ?fr3 snvoc:knows/snvoc:hasPerson ?fr. }
    }
    FILTER (?fr != sn:pers00000024189255811103).
```

path length 1

path length 2

path length 3

Graph path search: Querying ALL PATHS

- GraphDB Path Search **service extension** where path search parameters are specified
- Traversal with both **Source & Destination** nodes or just one of them
- **Path bindings** and result displaying
- Optional **Min & Max Path Length** params
- Graph Pattern Nested Service specifying the **relationship between nodes** in path

```
SELECT ?fr ?indexInPath ?pathNo WHERE {
    BIND (sn:pers00000024189255811103 as ?source)
    SERVICE path:search {
        <urn:path> path:findPath path:allPaths ;
                   path:sourceNode ?source ;
                   path:destinationNode ?destination ;
                   path:resultBindingIndex ?indexInPath ;
                   path:pathIndex ?pathNo ;
                   path:maxPathLength 3;
                   path:startNode ?start ;
                   path:endNode ?fr .
        SERVICE <urn:path> {
            ?start snvoc:knows/snvoc:hasPerson ?fr.
        }
    }
}
```

Graph path search: Querying SHORTEST PATH

- **Shortest Path** function
- **Wildcard** pattern where adjacent resources are evaluated on a level-by-level basis allowing the edge label to be exported as well
- **Directional** (by default) as well as **bidirectional** search

```
SELECT ?edge ?indexInPath ?pathIndex WHERE {
    VALUES (?src ?dst) {
        ( sn:pers00000024189255811103 sn:pers00000021990232601 68 )
    }
    SERVICE path:search {
        <urn:path> path:findPath path:shortestPath ;
                   path:sourceNode ?src ;
                   path:destinationNode ?dst ;
                   path:resultBindingIndex ?indexInPath ;
                   path:pathIndex ?pathIndex ;
                   path:resultBinding ?edge ;
                   path:bidirectional true ;
    }
}
```

Graph Path Search in RDF Databases

	SPARQL 1.1 compliant syntax	Complex pattern and relations	Distinct paths	Wildcard predicates
GraphDB	Y	Y	Y	Y
Stardog	N	Y	Y	Y
BlazeGraph	Y	Y	N	Y
AllegroGraph	Y	N	N	Y
AnzoGraph	Y	N	N	N

Note: Neptune and ORACLE do not offer SPARQL extensions for graph path search

- Intro & Ecosystem
- Essentials
- High Availability & Security
- Workbench & Visualizations
- Search & Analytics
- Reasoning & Validation
- Edge Properties, RDF-star
- Graph Path Search
- Geographic Query Extensions
- Benchmarking
- Latest Releases

ontotext

Geographic Data Extensions

Geospatial Plugin:

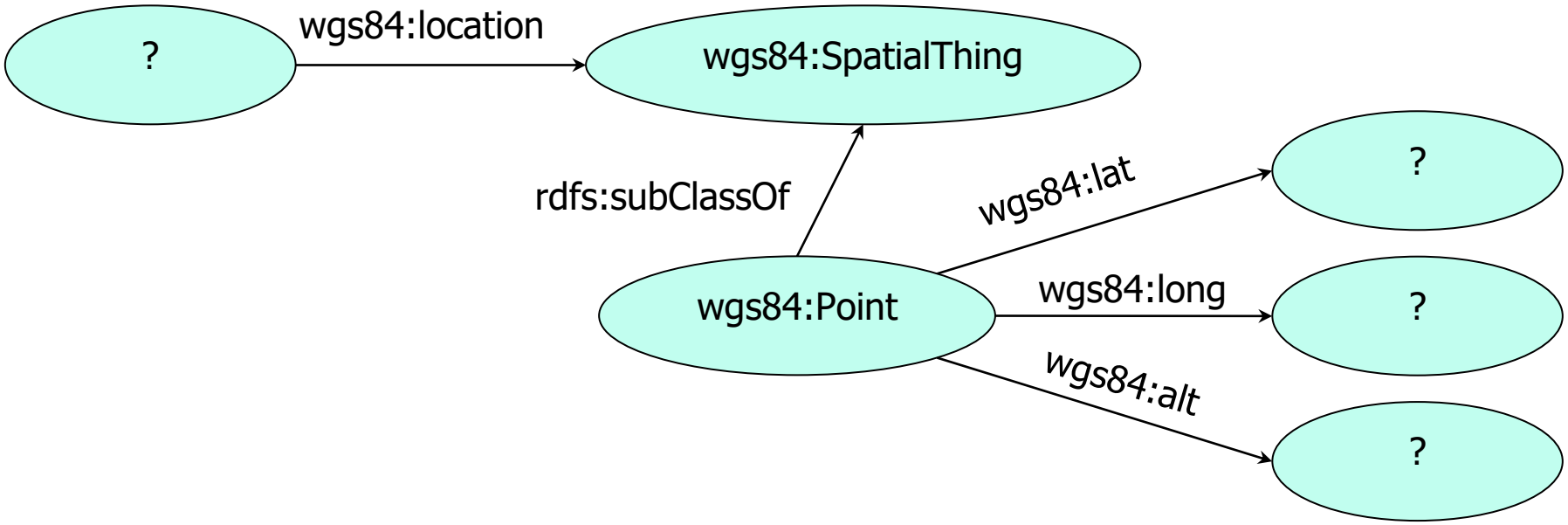
- Special handling of 2D geo-spatial data using a dedicated index
- Provides query constraints, otherwise not possible in SPARQL
- Fast and efficient; based on R-Trees

GeoSPARQL Support:

- OGC standard for representation and querying of geographic linked data
- GraphDB supports GeoSPARQL 1.0

Geospatial Plugin Data Assumptions

- **Data must use the WGS84 ontology**
  - http://www.w3.org/2003/01/geo/wgs84_pos



Geospatial Plugin Features

- **Provides a set of magic properties, e.g.**

  ?link **omgeo:nearby**(?lat1 ?long1 "50mi")

  This is interpreted as:

  ?link geo-pos:lat ?link_lat .

  ?link geo-pos:long ?link_long .

  where the distance between (?link_lat, ?link_long) and (?lat1, ?long1) is less than or equal to 50 miles

- **Query extensions include:**
  - Nearby (lat long distance)
  - Within (rectangle)
  - Within (polygon)
  - distance function

Geospatial Plugin Features (ctd)

- **e.g. tunnels lying within a rectangle enclosing Tirol:**

```
PREFIX geo-pos: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX geo-ont: <http://www.geonames.org/ontology#>
PREFIX omgeo:   <http://www.ontotext.com/owlim/geo#>
SELECT ?feature
WHERE {
  ?link omgeo:within(45.85 9.15 48.61 13.18) .
  ?link geo-ont:featureCode geo-ont:R.TNL .
  ?link geo-ont:name ?feature .
}
```

- **e.g. airports near Seoul**

```
SELECT distinct ?airport
WHERE {
   ?base geo-ont:name "Seoul" .
   ?base geo-pos:lat ?latBase ; geo-pos:long ?longBase .
   ?link omgeo:nearby(?latBase ?longBase "50mi") .
   ?link geo-ont:name ?airport ;
   ?link geo-ont:featureCode geo-ont:S.AIRP .
}
```

GeoSPARQL Support

- **Represents geographic data using a small topological RDF/OWL ontology**
  - Geometry literals encoded as GML or WKT
- **Defines vocabulary to query various topological relations**
  - OGC Simple Features
  - Region Connection Calculus (RCC8)
  - Dimensionally Extended 9-Intersection Model (DE-9IM), also known as Egenhofer/Clementini relations
- **Provides various non-topological query functions**
  - For example: distance, intersection, union, difference
- **Backed in GraphDB by a dedicated Lucene-based index**

- Intro & Ecosystem
- Essentials
- High Availability & Security
- Workbench & Visualizations
- Search & Analytics
- Reasoning & Validation
- Edge Properties, RDF-star
- Graph Path Search
- Geographic Query Extensions
- Benchmarking
- Latest Releases

ontotext

GraphDB Benchmarking

- **Linked Data Benchmarking Council: TPC-like industry body for graph databases**
  - ✓ Members: Ontotext, Neo4j, AWS, ORACLE, TigerGraph, CWI, UPM, *Sparsity, ….
- **GraphDB results on SPB and SNB were audited Q4'22 by the Linked Data Benchmarking Council**
  - ✓ neo4j designed SNB with accent on graph traversal and analytics, but never published results form it
  - ✓ GraphDB's results are the only ones where a general-purpose database engine passes the benchmark without custom indices or compression tailor made for this benchmark!

LDBC The graph & RDF benchmark reference

ontotext

Semantic Publishing Benchmark

- Replicates BBC's Dynamic **Semantic Publishing approach** through
  - BBC implemented this first for their FIFA World Cup website in 2010
  - **Large volume** of **streaming content**, e.g. creative works and media assets
  - **Enriching content** with **metadata** that describes it and links it to reference knowledge - information about entities: players, teams, groups, matches
  - **Regular updates to the metadata** and less often updates to the reference knowledge
  - Aggregation queries, that retrieve content according to various criteria
- Challenges multiple possible bottlenecks in engine performance (full scans etc.)
- Combines **frequent updates** with **inference**, **geospatial constraints** and **FTS**

GraphDB combines high-availability and scalability on SPB

Audited benchmarks runs:

- Two scale factors:
  - Scale factor 5 (SF5, **SPB 1B**) – a graph of 1.4B edges, after inference materialization
  - Scale factor 3 (**SPB 256M**) - a graph of 400M edges, after inference materialization
- Workload: **Aggregation agents** (12 queries) + **Editorial agents** (2 updates queries)
- Hardware: **AWS r6id.8xlarge** server (256GiB RAM, 32 vCPUs, Intel Xeon 8375C)
- Two configurations: **single server** and high-availability replication **cluster of 3 nodes**

Further reading: SPB main page with audited results: https://ldbcouncil.org/benchmarks/spb/

Cloud-ready graph database within minutes

- **Quick & easy to set up and replicate benchmark results**
  - Available helm chart and docker images both for single instance and clustered setups
  - No parameter tunings required to match performance from audited results
- **Scaling performance with cloud hardware**

AWS Cloud Instance	Cost/ hour	SPB 1B 16 read agents (QPS)
i4i.4xlarge	$0.89	**72**
m6id.8xlarge	$1.20	**104**
r6id.8xlarge	$1.52	**130**

Notes on the table:

- *Unaudited data in the table above; QPS=Queries per sec.*
- *GraphDB Single instance with 80 GiB heap*
- *Cost for 1Yr reserved as of June 2023*

Read/Write Agents	SPB 256M R/W Ops	SPB 1B R/W Ops
0/4	0/38	0/17
8/4	217/31	69/13
16/4	335/26	106/10
24/0	413/0	158/0

Hardware: r6id.8xlarge

ontotext

Enterprise grade graph database within minutes

Scaling throughput with # of concurrent users in a cluster

Read/Write agents	SPB 256M QPS (Queries per Second)	SPB 1B QPS (Queries per Second)
16/0	467	181
32/0	755	305
64/0	986	409

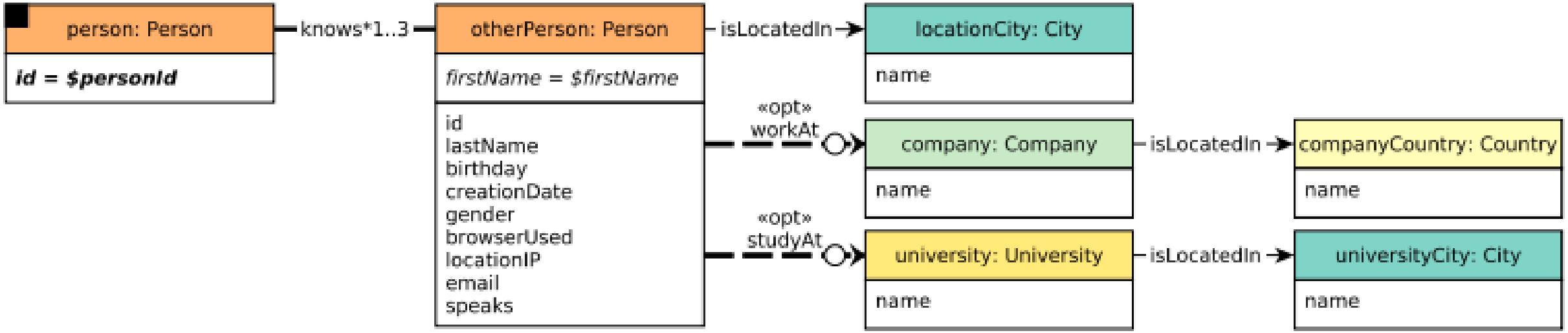
** GraphDB 3-node high-availability cluster, r6id.8xlarge instances with 32 vCPUs*

- **Serving ~1000 QPS to 64 clients** with high availability cluster
- **Effective query load balancing** across the nodes in the cluster
- Sublinear query performance for growing datasets

LDBC Social Network Benchmark

- **Benchmarking vs. just "checking the box"**
  - ✓ Test time & space complexity
  - ✓ Detect choke points and implementation inefficiencies
- **Linked Data Benchmarking Council: TPC-like body for graph databases**
  - ✓ Members: Ontotext, Neo4j, AWS, ORACLE, TigerGraph, CWI, UPM, *Sparsity, Katana, …
- **LDBC SNB is the most comprehensive graph analytics benchmark**
  - ✓ Analytics-oriented loads designed to simulate operations in a social network platform
  - ✓ Lots of research work invested in a sophisticated data generator, making sure the data distributions and connectivity are "good": both realistic and challenging

SNB person data model & sample queries



- People that a person is connected to at up to 3 steps via ***knows***
  - ✓ Part of **SNB Interactive Q1**
- Find the shortest path between two persons
  - ✓ Part of **SNB Interactive Q14**

Extending SNB compatibility to RDF & SPARQL

- **Implementing SNB Interactive driver**
  - **No imperative language** or stored procedures based query execution
- **GraphDB Path-search extension used for traversal**
  - Graph path search/traversal is very clumsy to implement in vanila SPARQL
  - GraphDB's Path search extension is **compliant with SPARQL 1.1** syntax, unlike other triplestores
- **Data loading with SNB Hadoop data generator**
  - Audited dataset generator with no modifications to data model

SNB Interactive Challenges

- Complex query plans with multiple JOINs and OPTIONAL clauses required
- Numerous aggregation queries and path traversals
- Multiple-hop queries matched with joins of related metadata
- Frequent data update queries
- Combined complex analytical with lightweight throughput queries

Ontotext's Approach

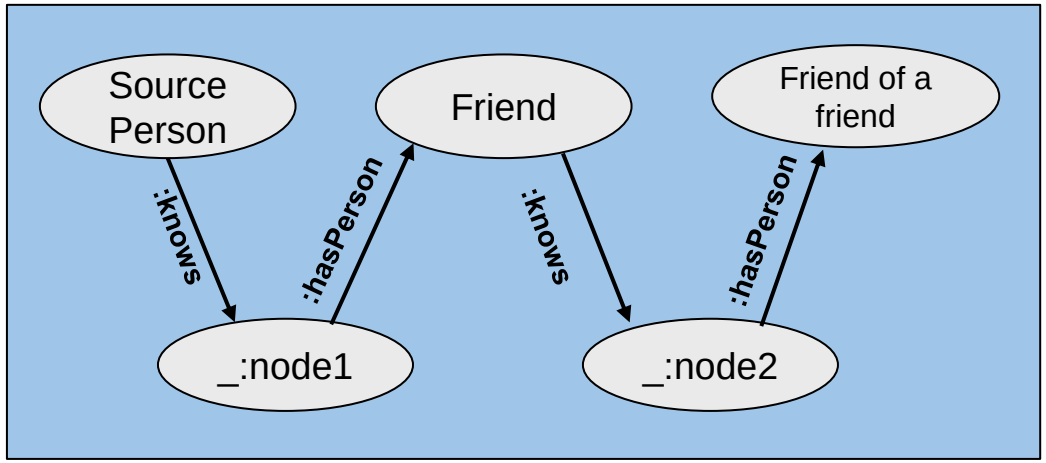
- Optimized All-path traversal memory utilization by leveraging our global entity pool
- Optimized Shortest-path traversal by implementing a greedy approach to iterate adjacency lists
- Used inference to materialize "shortcuts" in the graph

Queries SF10	**An LPG engine** (ms)	**GraphDB** (ms)	**AVG reads Δ base**
6: all path search	4,303.25	1,631.12	**-62.1%**
14: shortest path + weight	2,037.14	812.40	**-60.1%**

Optimizing query performance with inference

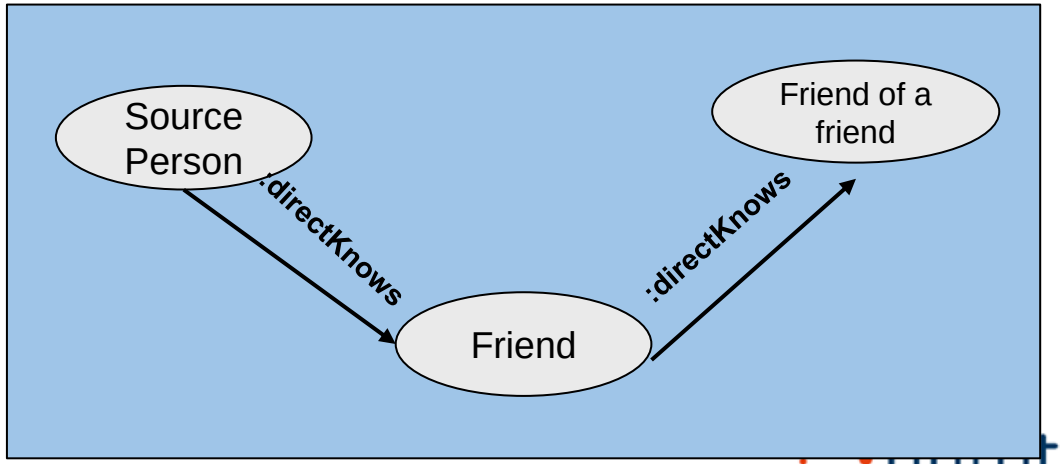
Shortest Path Extract - Q11 without Inference

```
SERVICE path:search {
    <urn:path> path:findPath onto:shortestPath ;
               path:sourceNode ?source ;
               path:destinationNode ?destination ;
               path:maxPathLength 2;
               path:startNode ?start ;
               path:endNode ?fr .
    SERVICE <urn:path> {
        ?start snvoc:knows/snvoc:hasPerson ?fr.
    }
}
```



Shortest Path Extract - Q11 with Inference **(25x times faster)**

```
SERVICE path:search {
    <urn:path> path:findPath onto:shortestPath ;
               path:sourceNode ?source ;
               path:destinationNode ?destination ;
               path:maxPathLength 2;
               path:startNode ?start ;
               path:endNode ?fr .
    SERVICE <urn:path> {
        ?start snvoc:directKnows ?fr.
    }
}
```



GraphDB: The First RDF Engine to Pass SNB

Audited results:

- Scale factor 30 (SF30) – a graph of 1.5 billion edges
- Workload: Interactive (14 queries)
- Hardware: AWS r6id.8xlarge server (256GiB RAM, Intel Xeon 8375C)
- **12 ops./sec.** on a driver configured with **4 read and 4 write threads**
  - Linear scalability - the result with single agent is 3 ops./sec.

The first audited result for system with declarative query language!

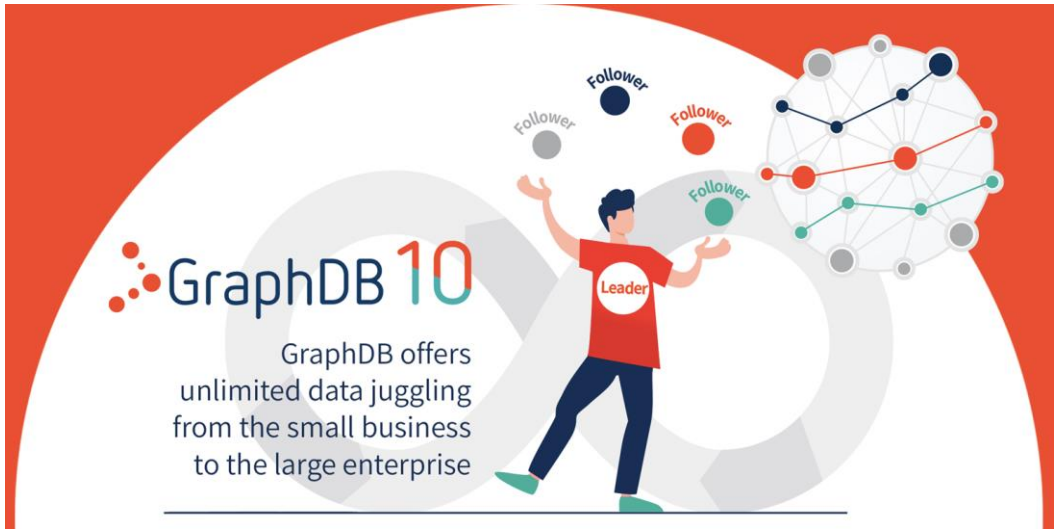
Further reading:
SNB main page: https://ldbcouncil.org/benchmarks/snb/
Audited results are published at https://ldbcouncil.org/benchmarks/snb-interactive/

ontotext

- Intro & Ecosystem
- Essentials
- High Availability & Security
- Workbench & Visualizations
- Search & Analytics
- Reasoning & Validation
- Edge Properties, RDF-star
- Graph Path Search
- Geographic Query Extensions
- Benchmarking
- Latest Releases

GraphDB 10.0

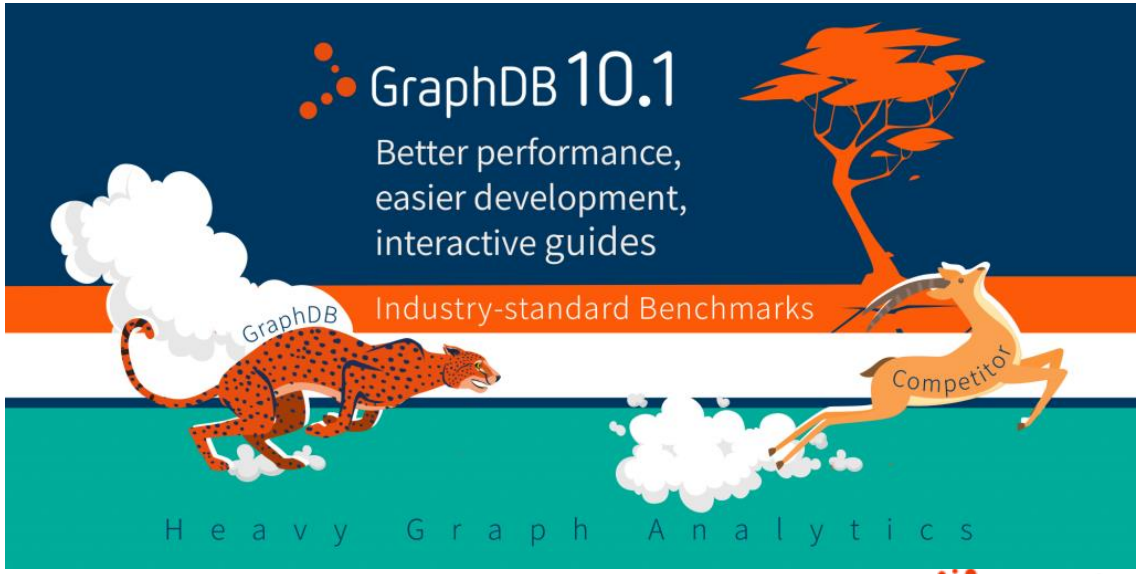
- New high-availability cluster
- Single distribution and repository type
- Licensing and parallelism
- Connector filtering redesign
- Upgraded to RDF4J
- Ontotext Refine removed from GraphDB



ontotext

GraphDB 10.1

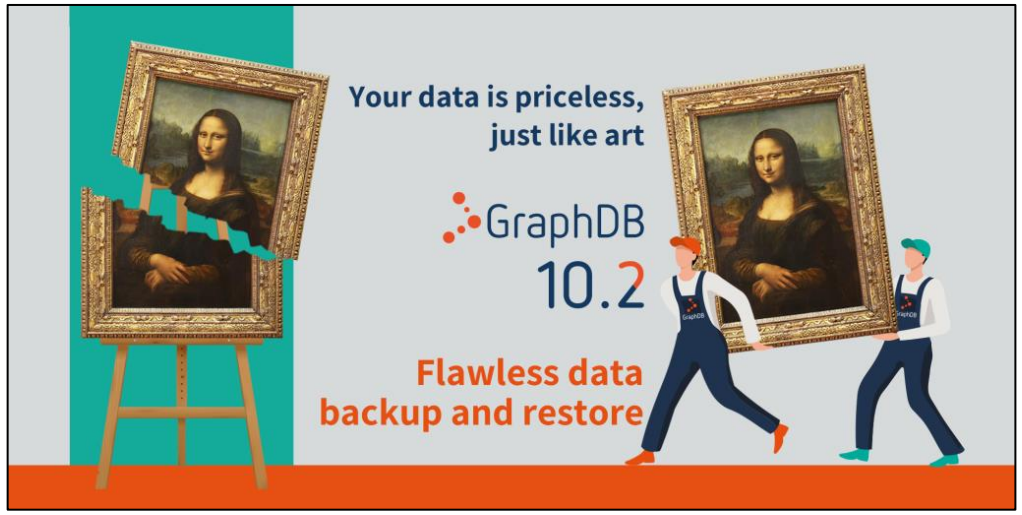
- Significant performance improvements and lower memory usage
- Easier start with the product
- New SPARQL function and better compatibility with Jena
- Simple full-text search
- Interactive user guides

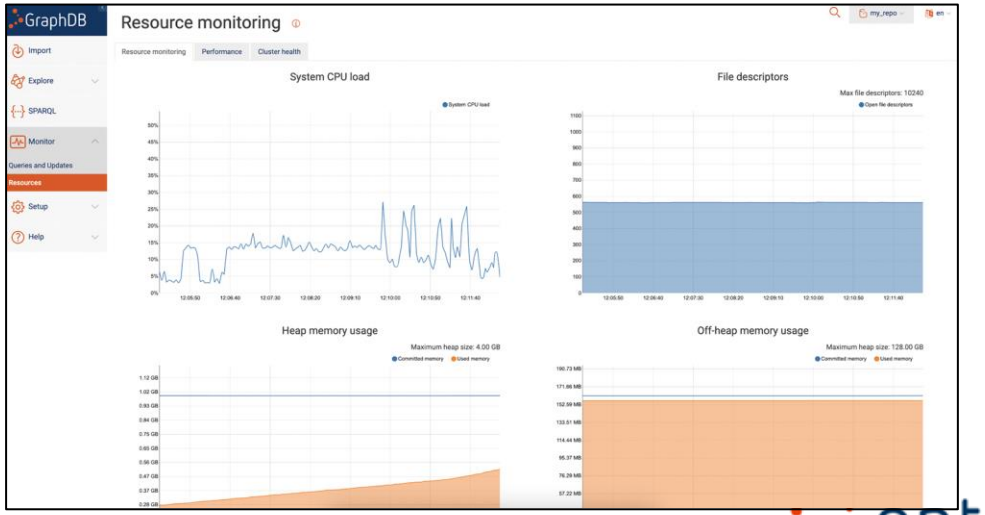


ontotext

GraphDB 10.2

- Improved cluster backup and support for cloud backups
- Lower memory requirements & improved transparency memory mode
- Better monitoring and support for Prometheus
- Flexible authentication options with X.509 certificate

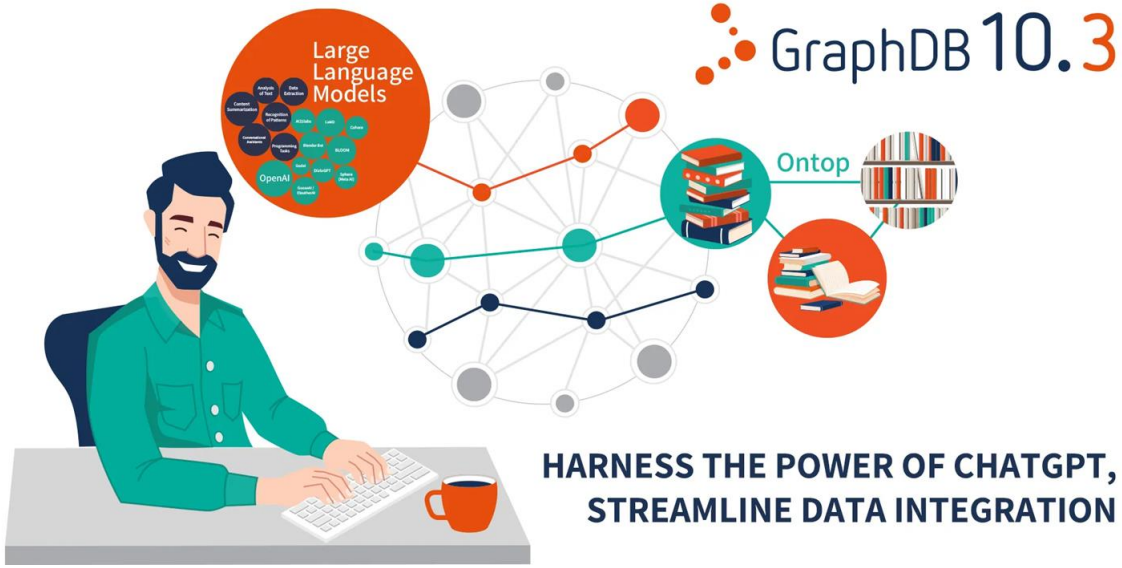


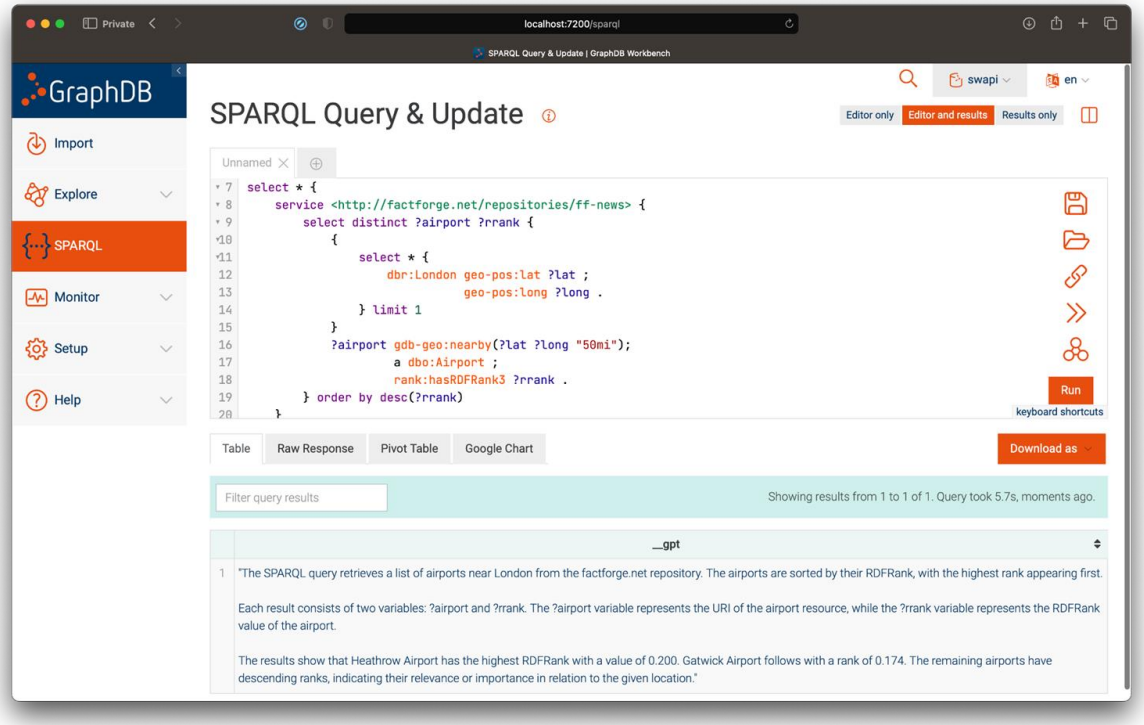


ontotext

GraphDB 10.3

- Seamless integration with Large Language Models
- Integration with data management platforms
- Connectors for the world





ontotext

GraphDB 10.4

- Integration with vector database/ChatGPT retrieval plugin
- Named graph and quad level security
- Publish GDB on the AWS Marketplace
- Detailed cluster monitoring API and view
- Users audit trail in the history plugin

GraphDB Roadmap 2023

Strategic themes	Key features
Ease of use and dev experience	Quad-based access control, improved UX, better documentation, language specific libraries with cluster discovery, JSON-LD 1.1, execute SPARQL template queries via SERVICE
Improve the performance	New in-memory repository type to handle big data efficiently
Integration with ML, LLMs and analytic algorithms	Use LLMs to answer questions about your KG
Integration with DM platforms	Increase the number of supported SQL sources
Easier consumption of knowledge graphs	Tighter integration with GraphQL
Cloud offering over AWS and Azure	Publish GraphDB on AWS/Azure marketplace
Security and database reliability	Address known vulnerabilities, backup compression, automatic transaction log size management

Product Roadmap on LLMs

- **Integration with Vector Databases**
  - Automatic transactional safe synchronization of vector databases
  - Select a vector space model compatible with LLMs
- **Grounding of LLM with KG via a vector database**
  - Develop examples of pattern “Grounding Context (Retrieval plugin)”
  - Evaluate the pro/cons of the grounding approaches
- **Expand the list of supported LLMs**
  - Avoid data privacy issues with local LLMs
- **LLM as an NLP tool**
  - Productize generic relation extraction with prompt engineering
  - Integrate with generic or optimized entity linking algorithms

Give It a Try!

Play with GraphDB at FactForge.net

Explore a public read-only access to a GraphDB repository loaded with 2 billion facts, incl. DBPedia, Geonames and metadata for 1 million news. An easy way to experiment with GraphDB functionalities such as SPARQL editor, visualization, FTS connectors, geo-spatial and ranking.

Download and use the single-click installation of GraphDB Free

Delve into all GraphDB functionalities, including OntoRefine, which allows a WYSIWYG transformation of tabular data and reconciliation.

Get an Evaluation License for GraphDB Enterprise Editions

ontotext

Data Pieces

Integration

Smooth Data

THANK YOU!

www.ontotext.com

sales@ontotext.com

ontotext