

# Semantic SPARQL queries: a novel federation model and implementation towards Enterprise Data Governance

Mirco Casoni, Stefano Monti,  
Francesco Sprotetto  
Imola Informatica s.p.a.  
Via Selice, 66/a  
40026 Imola, (BO), Italy  
{mcasoni, smonti,  
fsprotetti}@imolinfo.it

Antonio Corradi,  
Luca Foschini  
DISI – University of Bologna  
Viale del Risorgimento, 2  
40136 Bologna, Italy  
{antonio.corradi,  
luca.foschini}@unibo.it

Riccardo Venanzi  
Department of Engineering –  
University of Ferrara  
Via Saragat, 1  
44122 Ferrara, Italy  
riccardo.venanzi@unife.it  
riccardo.venanzi@unibo.it

## ABSTRACT

Large and complex organizations all over the world pose relevant challenges regarding Data Governance and Federation as result of the integration of heterogeneous, distributed data sources. The key in searching and defining semantic on any sort of content on the web is Semantic Web, with its most used language SPARQL. It easily allows the clients to discover hidden relationships among largely heterogeneous data. However, current SPARQL support for Data Federation presents some fair limitations, that make its use unfeasible in real-world scenarios. An open and autonomous platform for Data Federation is proposed in our work; it aims to overcome traditional SPARQL limitations and opens up remarkable opportunities for Data Governance in large organizations.

## CCS Concepts

- Information systems for semantic web description languages

## Keywords

Enterprise Data Governance; Federation; Semantic Web.

## 1. INTRODUCTION

Discovering, aggregating, and managing data in heterogeneous forms and from different and distributed sources are the relevant challenges posed by the governance of large distributed organizations. The governance models produced typically have a limited organization knowledge and data integration, with no flexibility. The case of the integration of Enterprise Resource Planning (ERP) systems data with both traditional Custom Relationship Management (CRM)/marketing analysis and with innovative social media analysis systems to proactively condition enterprise production and operations, e.g., supply chain and stock management tuning as well as workforce optimization, could be a classic example.

Modeling and managing complex relationships between entities/data, supporting interoperability, reasoning and inference/automation in coarse-grained, semi-structured, and

heterogeneous data environments, are activities that can be easily handled by exploiting semantic web principles, methodologies, and techniques: in this field, the SPARQL protocol (based on the Resource Description Framework – RDF – specification [1]) has been widely well recognized as the de-facto standard to extract and manipulate information from distributed data sources on the web [2, 3].

However, traditional Semantic Web methodologies and techniques typically exhibit severe limitations in large heterogeneous scenarios where multiple distributed data sources have to be integrated. In those cases, Semantic approaches would require complete *a priori* knowledge of all data sources and their network distribution and topology.

That limitation becomes particularly crucial in Enterprise IT/Data Governance scenarios where a complete, durable, and *a priori* data source knowledge is often unrealistic and infeasible. A typical example would be one where multiple public and private academic institutions are willing to integrate to optimize their geographic and demographic coverage (e.g., to avoid geographic course overlap from different academia). Such synergic education offering would require a continuous data integration flow from both education (e.g., currently available courses and teaching areas) and administrative departments (e.g., students distribution, cost and revenue distribution per area and department), with demographic data from municipalities (e.g. Open Data about citizen distribution per geographic area and age).

Both municipalities and academic institutions will likely feature their own, very different and constrained data sources, information systems and data gathering and management processes, so making them converge to single, uniform and standardized data sources, systems, and processes would be simply unrealistic. Similarly, both academic institutions and municipalities should be left free to change/evolve their own IT infrastructures/systems and processes, and still guarantee data interoperability and integration.

Despite the aforementioned limitations in terms of distributed data source knowledge, we strongly believe that flexibility and extensibility of Semantic approaches are particularly suitable for IT/Data Governance scenarios such as the ones described above. An innovative semantic-based approach to overcome organization governance complexity and to mitigate/hide heterogeneity and distribution of data sources is presented in this work: our paper describes a reference architecture model that relies on a federation of SPARQL endpoints, as well as a full implementation on top of an existing, enterprise-grade largely adopted Semantic framework, and real use-case scenarios to demonstrate the feasibility of our proposal.

Our approach overcomes the classic governance models (where integration and data harvesting are difficult, expensive, and too

often related to specific domains/areas), and eases an horizontal, much broader-scoped, and continuous integration of data: a higher level of data normalization and the integration is promoted by our approach, in addition, a more proactive governance model where data analysis may be used not only to lead retrospective analysis, but also to proactively support upcoming strategic decisions is proposed. We are going to use the above integration example as a fil rouge throughout the paper.

Section 2 describes related work in the area of both Data Governance and Semantic approaches and Section 3 defines the conceptual model of our solution. Section 4 and 5 provide relevant insights in both design choices and implementation details of our work, and Section 6 describes some empirical evaluations to prove the feasibility of our approach. Finally, Section 7 includes conclusions and future work we are planning to carry on this topic.

## 2. RELATED WORK

Companies have long recognized information systems as important assets. Nowadays, methodologies and techniques for managing such crucial IT assets are strongly driven by IT and Data Governance disciplines.

One of the most relevant decision-making processes is IT Governance, in addition it supports the dynamic adaptation to ever-changing business/market needs and goal achievement. In particular, IT Governance deals with Resource Management, Strategic Alignment, Delivering Value, Risk Management, and Performance Measurement [4, 5]. While, the convergence of multiple functionalities such as Data Quality, Data Management, Risk Management, and Policies definition and enforcement, are dealt by Data Governance [6, 7].

Understanding the data flows across an organization, which are the constraints that must be considered, which is the regulatory landscape, who is responsible for the data and who can use the data themselves, are activities related to Data Governance's area. Once these topics have been clearly shaped, rules on the data model and the data itself are defined by the Data Governance.

In the synergic academic offering example we depicted in Section 1, a Data Governance initiative in such a distributed ecosystem would for instance:

- identify data sources among academia/municipalities information systems;
- identify data responsibility and ownership: e.g., information about current students involved in courses may come from both education and administrative departments, and deciding which one should be considered master in case of disputes is key;
- identify private, non-shareable data and enforce data privacy: e.g., personal students information such as age, gender and personal contact information should be provided only as aggregate data;
- guarantee a minimum set of data to be provided by each data source (e.g., each academic institution should at least provide student distribution per age and geographic site).

In Data Governance, a crucial role is played by Data federation, understood as the ability to seamlessly and transparently integrate data stored at different places. Data federation is usually seen as a specialization of Data integration systems, where data storage leverages fully integrated, fully distributed models, abandoning the idea of single centralized physical endpoints.

In the above example, Data federation guarantees transparent data interoperability and integration no matter the actual information system technology and vendor (e.g., depending on the specific

academic institution, student distribution may reside on databases of different vendors, on legacy CSV files, as well as proprietary ERP systems data formats).

Semantic Web has long ago came out as tools for enriching contents (HTML pages, binary files, images/media content, and so on) in the World Wide Web with metadata that specify the semantic context of any content itself. Metadata formats and languages (such as RDF and OWL) are mainly designed to efficiently express content information, and automatically execute semantic data query (e.g., via search engines), interpretation, and, more commonly, to automatic aggregation and reasoning. One step further between computers and humans is advanced by Semantic Web, and it allows humans to exploit a more intelligent and autonomous machine support in the execution. W3C SPARQL is a key element of the Semantic Web, it is a query language for Resource Description Framework (RDF), and has long set itself as the de facto standard to execute semantic queries on Web content. RDF defines the concepts and relationships about the Web content introducing triples (subject-predicate-object) for describing data; A knowledge graph is created with the triples that have some elements in common. The navigation of such knowledge graphs and the searching for sub-graphs corresponding to user requests are driven by SPARQL. Data Governance and federation needs are supported by Semantic Web and SPARQL query language providing a promising platform that also allows to build a connected network of information [8]. However, federation of semantic data and navigation via SPARQL queries is still at an early stage and requires users to explicitly express the distributed nodes upon which to perform semantic queries and subsequent result aggregations, therefore negating the intrinsic benefits of adopting a semantic approach to distributed data aggregation and reasoning [9, 10].

## 3. THE SPARQL FEDERATION MODEL

Information flows are easily and effectively shaped and supported by the organization across its branches through Data Federation. An effective and viable model and implementation for adopting Semantic Web methodologies and the SPARQL implementation is defined by our solution, that aims to overcome typical issues such as heterogeneity and distribution of data sources.

Key principles in defining our solution are

- *Openness*: integrating data sources from heterogeneous (both custom and third party) systems is what the large organizations means for data federation and integration; preserving openness, portability and avoiding vendor lock-in is the best way for defining a sustainable, long-term data federation strategy for any organization;
- *Lightness*: one of the main goal of the proposed solution is to avoid the overhead on running systems and minimize the impact of Data Federation on large organizations with complex, highly distributed data sources;
- *Autonomy and ease of use*: the proposed solution should be able to face with ambiguity, and to autonomously detect relationships among federated data even in case of partial a priori data model and network knowledge.

The key for enabling open, autonomous, machine-driven content matching and reasoning knowledge management infrastructures has long been proven to be Semantic Web standards and SPARQL query language. Nevertheless, the SPARQL support for data federation – via the SERVICE construct (e.g. distributed data source integration and query) – is still untimely and it is still poorly applicable to large, real-world scenarios. The limitations of current SPARQL data federation are related to the fact that a

complete, a priori knowledge of data models and actual data across all data sources are required for designing federated queries; In large complex scenarios where data sources are heterogeneous and can change frequently and at different paces from each other, this clearly becomes a relevant and strong constraint. The only way to surmount that constraint is to identify a new mechanism able to dynamically change its behavior in order to adapt itself to any scenario change, very likely to occur in large environments with many actors.

#### 4. ARCHITECTURE

This section addresses the three main architectural options we have evaluated for realizing a federation of SPARQL endpoints extensible and fully scalable. Several qualitative Key Performance Indicators (KPIs) have been defined in order to help us to compare and establish benefits and weak points of these alternatives, and to ultimately simplify the choice of the proposed solution. Any proposed are tested accordingly to those KPIs in order to compare their properties and ease the choice.

##### 4.1 KPIs

**Ease of development.** Our SPARQL federation, independently from the specific Semantic framework implementation, should require limited development efforts.

**Integration.** Other framework features should be easily integrated in our SPARQL federation solution, with minimum or no additional effort.

**Maintenance.** Our SPARQL federation solution should be thought for having the impact in terms of bug fixing or feature adaptation activities.

**Evolution.** Our solution should be easily extensible and open to new requirements.

##### 4.2 Alternatives

**Endpoint extension.** This solution is an extension of the internal logic of a SPARQL endpoint, it transforms the classic logic of the query execution for extracting data, into a model for the interception of the query, the query should be rewritten through the wired application logic, by adding the statement needed to query all the nodes in the federation, and finally execute on the numerous endpoints, returning the result in accordance with the provisions of the specific SERVICE clause SPARQL 1.1.

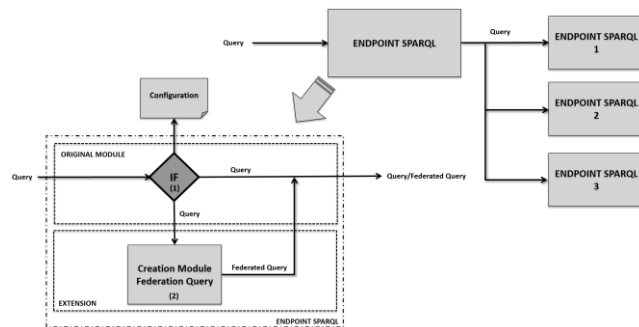


Figure. 1 – Endpoint extension alternative

This type of solution grants to provide input to any kind of SPARQL query, with the limitation of not being able to use the SERVICE clause, both because the manipulation uses that as the main construct, either because it would make the handling of the final query highly complex. This condition is relevant as it does not permit the use of the entire set of constructs that conform to

the standard SPARQL 1.1, giving the value added to the product that more to the solution.

The difficulty in handling the query itself is another aspect to consider, it could bring to not-trivial implementation of the query and poor performance. Indeed this approach is strongly dependent by the platform from both a management point of view (modifying source code from other vendors and distributing it may pose legal and organizational challenges in terms of distribution process and code ownership), and a technical one (need to modify existing SPARQL endpoint source code), hence obtaining a low grade on each KPI.

**Plugin.** Extending platform features via plugin modules is typically allowed in some Semantic platforms and SPARQL implementations. A dedicated plugin may realize Data federation and transparently handles data federation across nodes, overcoming current SPARQL constraints.

This approach presents non-trivial technology issues:

- the implementation of the plugin is strictly dependent by the actual semantic platform, this forces to federate only the nodes having the same semantic platform
- not the entire set of currently available, production-grade semantic platforms support a plugin model.

These constraints have a bad impact on all KPIs, only in controlled environments where the semantic platform is shared across the federation and a plugin model is supported, this solution is applicable.

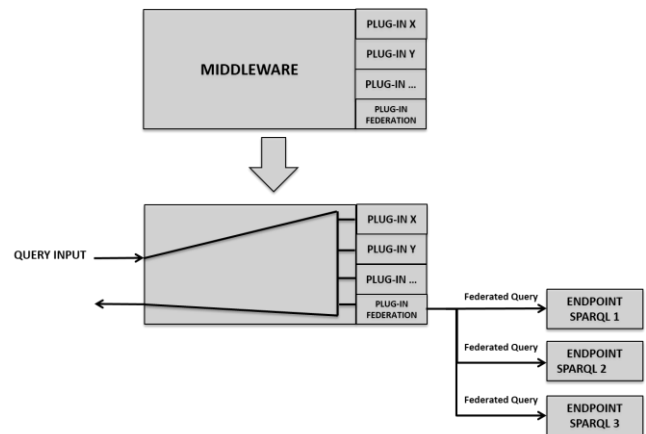
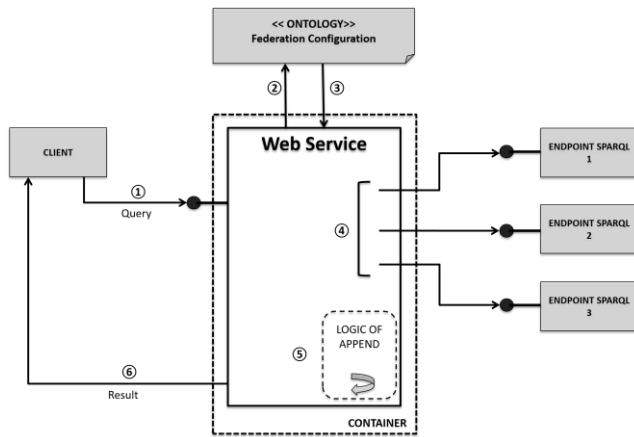


Figure. 2 - Plugin alternative

**Federation Web Service.** The creation of a Web Service is the base idea of the Federation Web Service solution; the service will realize all the federation-related functions. This solution is made portable to any semantic platform implementation by the Web Service approach, which also provides openness and interoperability.

Furthermore, we developed a specific Network Federation Ontology in order to facilitate the management of large semantic data networks and ease the definition and navigation of network topology.

The Federation Web Service, based on such ontology, transparently detects which endpoints and node should be involved in the query, facilitating the definition of federated semantic queries.



**Figure. 3 – Federation Web Service alternative**

The logical flow of the Federation Web Service is as follows:

- Federation resolution: The Federation Web Service is able to determine the current endpoints involved, through the Federation Ontology;
- Query execution: the federated query is performed on all individual nodes by the Federation Web Service;
- Result aggregation: the results obtained from single nodes are aggregated by the Federation Web Service, and returned to the caller. The typical constructs of the SPARQL language can easily manage the aggregation techniques, for example with the UNION clause.

#### 4.3 KPI evaluation and choice

**Table 1. Alternatives comparison**

	Endpoint extension	Plugin development	Federation WS
Ease of dev.	LOW - Platform-dependent	LOW Platform-dependent	HIGH
Integration	LOW - requires distributing a new version of the endpoint	LOW Platform-dependent	HIGH – no vendor lock-in
Maintenance	LOW - Platform-dependent	LOW Platform-dependent	HIGH
Evolution	LOW - Platform-dependent	LOW Platform-dependent	HIGH – no vendor lock-in

A high KPI score for this solution is granted by the openness and portability, mostly considering the points of view of development (ease) and management (integration, maintenance, evolution).

Our architectural choice therefore relies on the definition and implementation of a Federation Web Service.

## 5. IMPLEMENTATION DETAILS

This section highlights some relevant implementation details related to the selection of the Semantic middleware for our proposal (and related evaluation KPIs) and Federation Web Service implementation.

### 5.1 Middleware selection

The first step in realizing our SPARQL federation model relates to the selection of a Semantic middleware platform on top of which to build our solution.

We evaluated four main, widely-diffused Semantic middleware solutions, again on the base of a set of relevant qualitative KPIs.

**Apache JENA** [11]. Apache JENA is one of the most widespread, open-source semantic middleware on the market today. It is supported by the Apache community and is written entirely in Java programming language. JENA does not natively support SPARQL and SPARQL endpoints, though specific external libraries exist that provide both features (namely, the ARQ package [12] and the Joseki extension).

**Sesame** [13]. Sesame is an open source, Java-based RDF storage that natively supports SPARQL and exposes REST services to facilitate integration with external systems and services.

**Allegrograph** [14]. Allegrograph is an open source, Java-based RDF storage that natively supports SPARQL and exposes REST services to facilitate integration with external systems and services. SPARQL is the default query language in Allegrograph, and a implementation of the TWINQL [15] specification is also available; this specification provides relevant query optimization features.

**Openlink Virtuoso** [16]. Virtuoso is a multi-model data storage solution that combines traditional RDBMS storage features, with advanced Semantic storage, reasoning support (e.g., RDF), and an extensive set of integration features (e.g., REST/WebServices integration). SPARQL support is native in Virtuoso.

#### 5.1.1 KPIs

KPIs provided in the following are qualitative evaluation criteria relevant for any middleware choice. Our scoring attribution is based on the Capability Maturity Model Integration framework (CMMI), that provides a common reference model to assess the maturity of a system with respect to a given evaluation aspect [17].

Each KPI is evaluated against a set of values between 0 (no feature/characteristic support) to 5 (full, enterprise-grade feature/characteristic support).

**Federation support.** This KPI relates to the maturity of federation features provided by the candidate framework, and is the most relevant KPI in our evaluation model (i.e., highest weight).

Jena, Virtuoso, and Allegrograph provide no support for federation besides the limited SPARQL SERVICE directive, hence scoring an evaluation of 0 (no support) for this KPI. Sesame, instead, supports some sort of data sources federation, but data model sharing is still at an early stage, hence scoring an evaluation of 1.

**Community support.** This KPI relates to the maturity of the community involved in the development and maintenance of the candidate framework. Typical community support features relate to bug tracking and resolution processes, documentation, wiki, online support, etc, and are crucial for long-term manageability and stability of the platform. Allegrograph has virtually no active community and provides only a static documentation of the platform, hence scoring a lowest result. Jena and Sesame provide a fairly active community that involves bug tracking/resolution, and product roadmap definition. Virtuoso features the most active

community and provides bug tracking and resolution, product roadmap definition and direct on-call (chat) support.

**Commercial support.** Besides community support, commercial support is generally required in enterprise-grade software solutions to guarantee long-term stability. Commercial support typically provides highly skilled, special-purpose consulting resources as well as the ability to provide specific developments/extension and integrations.

Jena has no commercial support, Allegrograph and Sesame provide some level of consulting support, and Virtuoso provides the most comprehensive commercial ecosystem that offers also specific developments by need.

**SPARQL Endpoint support.** This KPI describes the availability of SPARQL endpoint APIs within the candidate middleware.

Virtuoso provides full endpoint exposition and support, whereas other candidates have only limited functionalities.

**Java support.** This KPI relates to product compatibility with the Java programming language. This KPI is particularly relevant due to the widespread adoption of Java (and its enterprise counterpart JEE) in developing large enterprise applications.

Jena implementation is fully based on the Java programming language, and exposes fully workable Java APIs; on the contrary Virtuoso is entirely written in C# and has limited API support for Java.

**Market adoption.** This KPI describes the availability skilled professional resources on the candidate framework. This KPI is especially relevant to guarantee manageability and evolvability of our solution. Virtuoso and Jena are leaders in terms of market adoption, hence featuring a larger market workforce share.

**Product maturity.** This KPI defines the overall maturity of the product, specifically in terms release process: products with frequent and long-term planned release schedules and roadmaps tend to be more reliable from a business point of view.

### 5.1.2 Middleware choice

This section synthesizes KPIs as described above, for each candidate middleware. Our evaluation is based on a weighted sum of KPIs, where the weight distribution reflects KPI relevance in our vision: federation support and market adoption are particularly crucial in our evaluation.

Table 2 details both KPI weight distribution (values ranging from 0 to 1) and scores for each KPI and middleware platform.

Aggregate results are obtained as a weighted sum of all KPIs: despite its poor Java support and lack of native federation support, Openlink Virtuoso is the Semantic platform that best fits our business needs and provides a stable and reliable option.

**Table 2. Semantic middleware comparison**

	Weight	Jena	Allegrograph	Virtuoso	Sesame
Federation support	0.3	0	0	0	1
Community	0.1	4	2	5	4
Commercial support	0.1	0	2	5	3
SPARQL endpoint support	0.1	1	2	5	1
Java prog. language	0.1	4	2	1	3

Market adoption	0.2	3	1	3	1
Product maturity	0.1	3	2	4	2
	1	1.8	1.2	2.6	1.8

## 5.2 Federation web service implementation

The implementation of federation has two key aspects: the Federation Web Service and the Federation Ontology that together can allow the complete support needed in our project.

### 5.2.1 Federation Ontology

The need of managing large amounts of data is mostly stressed in enterprise contexts, where this data often shows certain properties such as heterogeneity and distribution of the data model. For some business processes such as Decision Making, Data Quality, and Data Management, this aggregation of heterogeneous data is crucial. This aggregation encourages the concept of federation, which provides a federative pact on which to build the federation itself.

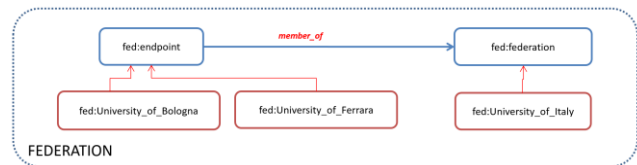
The implementation has followed these steps, firstly a uniform data model has defined and then shared among all member of the federation, thus the problems above described are eliminated.

The proposed ontology is composed by three main elements:

- the concept of federation
- the concept of the federation member
- the *member\_of* relationship that represents the relationship between the member of the federation and the federation itself.

The management of the federation is enabled by the ontology described above; this independently of semantic platform and it can be deployed and queried to any SPARQL endpoint.

From the point of view of the implementation, we realized our ontology according to the RDF schema modeling.



**Fig. 4 – Federation Ontology**

Figure 4 describes an implementation of our ontology that maps Italian academic institutions: *fed:University\_of\_Italy* represents a federation of universities, and *fed:University\_of\_Bologna* and *fed:University\_of\_Ferrara* are endpoint members of the *fed:University\_of\_Italy* federation.

### 5.2.2 Federation Web Service

The semantic data query and aggregation are managed by Federation Web Service, via the steps described in the following.

1. Clients can query the Web Service with a standard SPARQL query, and express a specific federation (in our examples, we modeled separate data sets from Italian universities)
2. The Web Service inspects the federation to determine which endpoints should be queried

3. The Web Service executes the semantic query on each such node
4. The results from the above queries are aggregated by the Web Service, then it returns them back to the caller

A particularly crucial task is result aggregation, and three main strategies are supported by our Federated Web Services:

- **INTERSECTION** strategy: this approach returns all the triples that are common to each involved endpoint;
- **UNION** strategy: results from endpoints are combined into a set of unique triples, hence performing duplicate detection and removal;
- **APPEND** strategy: results coming from different endpoints are simply stacked on top of each other, and sent back to the client once all replies are received; this strategy is implemented via the APPEND operator and does not handle result triple duplications.

The query definition, execution, and aggregation are clearly decoupled from the definition and management of the data set distribution by the Federation Web Service. Although highly distributed data sets can be handled by our solution, some management concerns may arise:

- **endpoint time-out**: how to handle alive nodes that fail to reply to queries in a given amount of time (e.g., due to temporary network issues); ignore/retry strategies should be carefully planned to reach a viable balance between result completeness (e.g. retrying queries in order to overcome the temporary network outage), and total response time;
- **endpoint unavailability**: endpoints may become unavailable for larger amounts of time, e.g. due to severe server/system failures.

## 6. EVALUATION

In this section we describe some Semantic platform benchmarks, as well as a specific use case of our SPARQL federation model.

### 6.1 Semantic platforms benchmarking

In order to prove the feasibility of our approach, we provide some useful insights about semantic platform performance under heavy load.

These quantitative analyses justify both Virtuoso adoption as the Semantic middleware/platform of our choice with respect to other contenders, and shed some light on absolute performance of Virtuoso under load.

**Dataset load time.** Loading large amounts of data becomes crucial in complex distributed scenarios where each node may contribute with a set of millions of records, thus limiting the overall performance of the whole system. The table below shows a load time comparison for datasets ranging from 1M to 100M entries, the times in the table are expressed according the dd-hh-mm-ss format.

**Table 3. Semantic middleware performance – dataset load time**

	1M	25M	100M
Sesame	00:02:59	12:17:05	3:06:27:25
<b>Virtuoso</b>	<b>00:00:34</b>	<b>00:17:15</b>	<b>1:03:53</b>
Allegrograph	00:00:52	00:36:49	00:48:30
Jena TDB	00:49	00:16:53	01:34:14

**Query execution.** Querying large amounts of data becomes crucial in complex distributed scenarios where each node may contribute with a set of millions of records, thus limiting the overall performance of the whole system. The table below shows a load time comparison for datasets ranging from 1M to 100M entries (Query per hour).

**Table 4. Semantic middleware performance – query execution/single client**

Single client	1M	25M	100M
Sesame	18094	1343	254
<b>Virtuoso</b>	<b>17424</b>	<b>12972</b>	<b>4407</b>
Allegrograph	4075	493	656
Jena TDB	4450	353	81

**Table 5. Semantic middleware performance – query execution/multiple clients**

Multi client	1M	25M	100M
Sesame	19057	18295	16517
<b>Virtuoso</b>	<b>28985</b>	<b>32668</b>	<b>33339</b>
Allegrograph	5861	7453	7888
Jena TDB	6752	8453	8664

The Semantic middleware we adopted for our reference implementation – Virtuoso – clearly outperforms its contenders both in terms of initial dataset load time, and in terms of query efficiency.

These numbers also clearly evidence how Virtuoso may scale linearly and handle extremely high workloads both in terms of initial dataset load, and in terms of single/parallel query executions, thus proving itself as a viable, production-grade option for large real data federation scenarios.

### 6.2 Use case

Our SPARQL Federation model allowed us to realize the complex integration scenario described in Section 1.

A proof of concept Federated Education Portal has been realized, it gives a synergic academic offering that federates education and administrative data sources from Italian academic institutions together with citizenship distribution data sources from Italian municipalities in order to create a more integrated education offering.

The actual data sources and their implementations are kept decoupled by a traditional Semantic approach, that allows to reuse the same SPARQL query over any academic and municipality data source, without caring of the implementation of real data (would it be an ERP system backed by a traditional RDBMS, or a legacy mainframe system).

However, a SPARQL-only approach for our Federated Education Portal would require to:

- have a priori knowledge of all municipalities and universities involved in the overall integrated offering;
- explicitly/manually execute the same SPARQL query on academic data sources to fetch students distribution;

- explicitly/manually execute the same SPARQL query on municipality data sources to fetch citizen distribution;
- explicitly map and combine both semantic result sets into a cohesive data set that highlights gaps in actual course offering with respect to real citizen distribution.

This process is surely poorly extensible and largely inefficient: any time a new data sourced is added, our Federated Education Portal should be extended, in order to query new endpoints and merge results with old ones.

Our approach leverages the following elements:

- A Federation Ontology implementation that maps Italian municipalities and relevant information about citizen geographic distribution;
- A Federation Ontology implementation that maps Italian academic institutions and relevant information about courses and student distribution;
- Students and Citizens are linked via the semantic notion of person (via the *foaf:Person* ontology);
- a set of Virtuoso instances as the default Semantic middleware;
- a set of SPARQL endpoints, for both municipalities and universities, on top of Virtuoso Semantic middleware
- an instance of our Federation Web Service.

In this scenario

- our Federated Education Portal executes a single query to fetch geographic distribution of both people involved in academic courses (students), and people (citizens) from municipalities;
- the Federation Web Service detects all involved academic and municipality data sources and transparently query each one of them;
- the Federation Web Service takes care of merging results (e.g., via an INTERSECTION strategy).

Our approach greatly simplify the realization of the Federated Education Portal:

- the Federated Education Portal does not require a priori hard-coded data source knowledge, resulting in a more open and flexible solution
- a single query can be executed both on municipality and on academia endpoints, hence making easier the development efforts of the overall solution.

## 7. CONCLUSIONS

In order to support ore Enterprise processes, a key role is played by discovery, aggregation, and manipulation of distributed, diverse sets of data. It has been proven that the semantic approaches validly infer the relationships and dependencies from heterogeneous sets of information pieces, however SPARQL - the current de facto standard query language - is not truly satisfactory when it has to perform a navigation in a real federated scenario with no exact knowledge of data distribution.

This work proposes a lightweight Federation Ontology and a Federation Web Service to map information sources across organizations, so as to address current SPARQL limitations in terms of a priori network knowledge.

Our Federated Web Service represents a portable, non vendor-specific solution that relies on the Federation Ontology to infer

network endpoints upon with to perform queries. The Federated Web Service then aggregates results via different composition strategies.

We adopted our Federation Web Service and Ontology to realize a Federated Education Portal that integrates data sets from both academic institutions and municipalities: our Federation Web Service and Ontology allowed us to dramatically ease the development of the portal itself. The system have been adopted by some of the involved organization as a reference for their internal projects.

We are currently investigating and testing strategies to overcome current limitations in terms of endpoint time-outs and unavailability.

## 8. REFERENCES

- [1] W3C Recommendation: *Semantic Web*. W3C, 2015 <http://www.w3.org/2001/sw/>
- [2] W3C Recommendation: *OWL Web Ontology Language*. W3C, 2004, <http://www.w3.org/TR/owl-ref/>
- [3] W3C Recommendation: "SPARQL 1.1 Federated Query", W3C, 2013, <http://www.w3.org/TR/2013/REC-sparql11-federated-query-20130321/>
- [4] D. Carstoiu; A. S. Cernian; V. Sgarciu; A. Olteanu, *The architecture of a distributed enterprise information management system based on Semantic Web*. 2010, Automation Quality and Testing Robotics (AQTR), 2010 IEEE International Conference on
- [5] L. Janahi, M. Griffiths, H. Al-Ammal. *A conceptual model for IT Governance: A case study research*, *Computer Vision and Image Analysis Applications*. (ICCVIA), 2015 International Conference on, Jan. 2015
- [6] P. P. Tallon. *Corporate Governance of Big Data: Perspectives on Value, Risk, and Cost*. IEEE Computer, Jun. 2013
- [7] T. Priebe, S. Markus. *Business information modeling: A methodology for data-intensive projects, data science and big data governance*. *Big Data (Big Data)*, 2015 IEEE International Conference on
- [8] G. Antunes, J. Borbinha, A. Caetano. *An Application of Semantic Techniques to the Analysis of Enterprise Architecture Models*. 2016, 49th Hawaii International Conference on System Sciences (HICSS)
- [9] N. A. Rakhmawati, M. Hausenblas. *On the Impact of Data Distribution in Federated SPARQL Queries*. 2012, *Semantic Computing (ICSC)*, 2012 IEEE Sixth International Conference on
- [10] A. Zimmermann; M. Pretz; G. Zimmermann; D. G. Firesmith; I. Petrov. *Towards Service-Oriented Enterprise Architectures for Big Data Applications in the Cloud*. 2013, 17th IEEE International Enterprise Distributed Object Computing Conference Workshops
- [11] APACHE Jena, <https://jena.apache.org/>
- [12] APACHE Jena ARQ, <https://jena.apache.org/documentation/query/>
- [13] Sesame, <http://rdf4j.org/>
- [14] Allegrograph, <http://franz.com/agraph/allegrograph/>
- [15] TwinQL extension, <http://www.cliki.net/twinql>
- [16] Openlink Virtuoso, <http://virtuoso.openlinksw.com/>
- [17] CMMI Model, <http://cmminstitute.com/>