ICT Seventh Framework Programme (ICT FP7)

Grant Agreement No: 318497

Data Intensive Techniques to Boost the Real – Time Performance of Global Agricultural Data Infrastructures

D1.3.1: Annual Public Report

Document Information	
Project Reference No.	ICT FP7 318497
Deliverable No.	D1.3.1
Relevant Work Package	WP1: Project Management; All Work Packages
Nature:	R
Dissemination Level:	PU
Document Version	v1.0
Date:	10/01/2014
Authors:	UAH, NCSR-D
Document description:	The present document provides an overview of the activities carried out and the overall progress achieved during the first year of the SemaGrow project.

1. Project Introduction

1.1 Project Vision

During the last years, the trend to open up data and provide them freely on the Internet has intensified in volume as well as quality and value of the data made available. The linked data community has grasped the opportunity to combine, cross-reference, and analyse unprecedented volumes of high-quality data and to build innovative applications. This effort has caused a tremendous network effect, adding value and creating new opportunities for everybody, including the original data providers.

But most of the low-hanging fruit has been picked and ***it is time to move on to the next step, combining, cross-indexing and, in general, making the best out of all public data***, regardless of their size, update rate, and schema; accepting that centrally-managed repositories (even distributed) are not able to meet the challenges ahead and that we need to develop the infrastructure for the efficient querying of large-scale federations of independently-managed sources.

1.2 Key Challenges

SemaGrow carries out fundamental databases research and develops methods and infrastructure that will be rigorously tested on three large-scale current use cases as well as on their projected data growth beyond project's end: we are laying the foundations for the scalable, efficient, and robust data services needed to take full advantage of the data-intensive and inter-disciplinary Science of 2020, by addressing the following key challenges:

- Develop ***novel algorithms and methods for querying distributed triple stores*** that can overcome the problems stemming from heterogeneity and from the fact that the distribution of data over nodes is not determined by the needs of better load balancing and more efficient resource discovery, but by data providers.
- Develop ***scalable and robust semantic indexing algorithms*** that can serve detailed and accurate data summaries and other data source annotations about extremely large datasets. Such annotations are crucial for distributed querying, as they support the decomposition of queries and the selection of the data sources which each query component will be directed to.

1.3 Summary of 1st Year Core Activities

The following table summarizes the core activities of the project undertaken during the 1st year of its implementation.

	Activity
1	Definition / Refinement of the overall architecture
2	First experiments on Resource Discovery
3	First experiments on Ontology Alignment
4	Initial Investigation of Content Classification & Ontology Evolution Techniques
5	Initial Investigation of Heterogeneous Distributed Semantic Querying Techniques
6	Setup of Semantic Store infrastructure for Large-Scale Experimentation
7	Initial definition of Scalability & Robustness Experimental Methodology
8	Definition / Refinement of Envisaged Applications & Use Cases
9	Initial definition of Real-life Deployment & User Evaluation Plan
10	Data Preparation

2. The SemaGrow Architecture

The core components that comprise the SemaGrow Stack, are depicted in the following Figure.

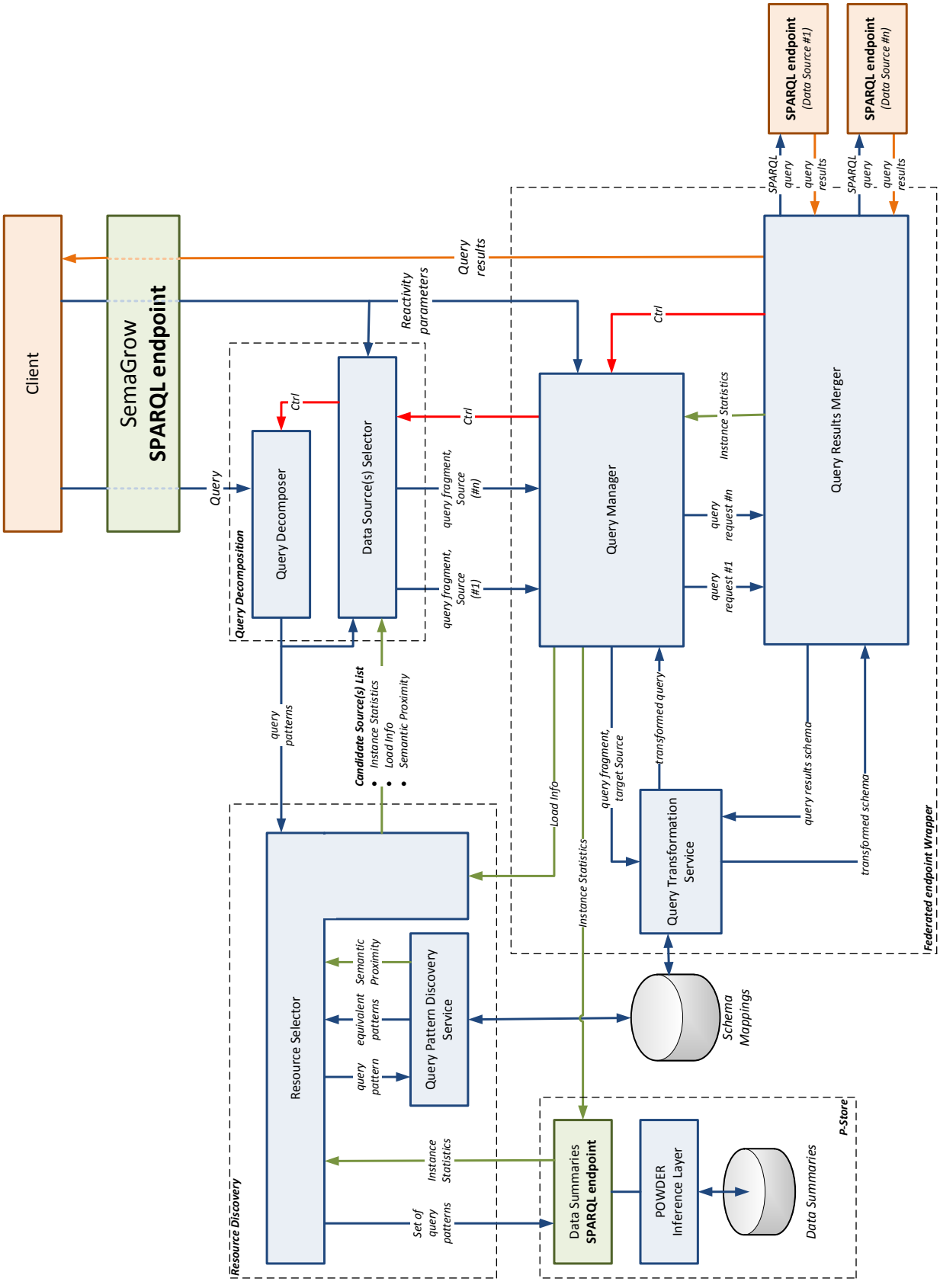

Figure 1: The SemaGrow Stack

The *SemaGrow Stack* integrates the components needed in order to offer a single SPARQL endpoint that federates a number of heterogeneous data sources, also exposed as SPARQL endpoints. The main difference between the *SemaGrow Stack* and most existing distributed querying solutions is that *SemaGrow* targets the federation of heterogeneous and independently provided data sources. In other words, *SemaGrow* aims to offer the most efficient distributed querying solution that can be achieved without controlling the way data is distributed between sources and, in general, without having the responsibility to centrally manage the data sources of the federation.

2.1 SemaGrow Stack Core Components

2.1.1 SemaGrow SPARQL endpoint

The *SemaGrow SPARQL endpoint* is the Web service through which client applications access SPARQL endpoints that have been federated using the *SemaGrow Stack*.

2.1.2 Query Decomposition

The *query decomposition* component analyses SPARQL queries and decides about the optimal way to break them up into query fragments to be dispatched to sources' endpoints. The *query decomposition* component comprises a *decomposer* module that syntactically analyses queries and suggests possible decompositions and a *selector* module that evaluates these suggestions using information and predictions from the *resource discovery* component about the data sources where each query fragment can be executed.

The result is a matching between query fragments and the source that each fragment is to be dispatched to. This is by necessity an approximation, since completeness can only be guaranteed by querying all sources. This is guided by the *reactivity parameters* that specify the client application's wished position in the trade-off between efficiency and completeness in terms of how much time it is possible and worth it to wait to get more results, or the minimum number of results required, or some other similar policy balancing between completeness and effort.

2.1.3 Resource Discovery

The *resource discovery* component provides an annotated list of candidate data sources that possibly hold triples matching a given query pattern; including sources that follow a different (but aligned) schema than that of the query pattern. The sources are annotated with schema and instance-level metadata and predicted response volume from the *data summaries endpoint*; as well as run-time information about current source load. When a source following an aligned schema is used, the annotation also includes relevant meta-information, such as the semantic proximity of the query schema and the source schema.

2.1.4 Data Summaries Endpoint

The *data summaries endpoint* serves metadata about the schema used and the instances stored in the various federated data stores. In particular, it indexes in the opposite direction than conventional data catalogues, by receiving entity URIs and responding with the repositories where triples involving these entities are located. Entities can be both at the schema level (classes, properties) and the instance level. Furthermore, it serves *ontology alignment* knowledge regarding entity equivalences between different sources. Although in principle any repository infrastructure can be used to store these data summaries, it is one of the core research objectives of *SemaGrow* to experiment with the POWDER protocol in order to take advantage of naming convention regularities to compress such indexes; and to develop a triple store that is especially well suited to serving metadata expressed using POWDER.

2.1.5 Federated Endpoint Wrapper

The *Federated End-point Wrapper* manages the communication with the external data sources that are federated by the *SemaGrow Stack*. Its *Query Manager* module is responsible for (a) where necessary, applying the *Query Transformation Service* to access repositories that follow a different schema than the one of the original query; (b) forwarding query

fragments to the *Query Results Merger*; and (c) collecting and forwarding dynamic run-time statistics to the *Resource Discovery* components.

The *Query Transformation Service* applies alignment knowledge served from the *schema mappings repository*. It re-writes query fragments from the schema of the original query to that of the data source that will be used for each fragment and also query results back into the schema of the original query so that they can be joined with results from other sources.

As joining distributed query results can degenerate into a situation where massive data volumes need to be copied to and processed by the results collector, *SemaGrow* envisages a *Query Results Merger* that exhibits pay-as-you-go behaviour, providing a first approximation with minimal usage of computational resources and iteratively refining it if more computation time and space are warranted by the *reactivity parameters* set by the client application. Distributed *incremental result fetching operators* exhibit this property, so that results can be incrementally requested and forwarded on arrival of new tuples. The confidence of mappings can be taken into account, offering higher priority to more certain mappings when needed, as per user requirements.

2.2 Maintenance Components

The system also foresees update and maintenance cycles, where new end-points are added to the federation or update the schema they employ or have accumulated considerable changes in the instances they hold. Schema metadata must be provided by the data provider when joining the federation, using authoring tools and tutorials produced by the project. Instance metadata may also be provided, but are also automatically maintained by the *resource discovery and query decomposition component* based on statistics extracted from query results.

2.2.1 Authoring Tool

SemaGrow develops a visual authoring tool that will assist data providers to author and maintain the data source annotations that are needed in order to take advantage of the full functionality of the *SemaGrow Stack*. This includes POWDER statements regarding the data and possibly other provenance and cataloguing meta-data.

2.2.2 Ontology Alignment Tool

The *ontology alignment tool* is responsible for aligning the various semantic vocabularies used by data providers and consumers. The component will present alignment suggestions to the data source providers to accept, modify, or reject. Accepted alignments will be recorded in the *data summaries repository* from where to be queried by the *SemaGrow Stack* components. The *ontology alignment tool* will assist data source providers to introduce new data sources, but also to integrate updates in the schema used in an existing source.

2.2.3 Content Classification and Ontology Evolution

Content classification and *ontology evolution tools* will also be used when new data sources are added. They will be used to refine coarsely annotated data and to bring annotations to a level of detail where they can be more accurately aligned with other schemas used in the federation.

3. 1st Year Experimentation Activities

3.1 Resource Discovery

The Resource Discovery component identifies the federated sources that are most likely to hold triples matching a given query and for breaking up queries in the fragments to be dispatched to each source for execution.

During the first year we focused on the development of a triple store that is capable of inferring triples from POWDER statements, as these provide a convenient (but hard to reason over) formalism for succinctly storing data summaries of what data is stored where in the federation.

We have chosen PostgreSQL as the physical layer for our POWDER store, due to its stability, efficiency, and extensive usage as triple store backend. Furthermore, it provides a built-in mechanism for customized indexing. Since POWDER inference is based on URIs' matching regular exceptions, we – effectively – need an efficient way of retrieving all tuples holding a value that matches a regular expression. We started with implementing list of tuple pointers for all tuples that match a single, fixed regular exception. Besides gaining familiarity with PostgreSQL indexing internals, this exercise has also shown us the maximum gain that we can ever hope to achieve: this is the perfect index for this single regular expression. As a second step, we extended this code so that instead of hard-wiring the regular expression, we now cache substrings vs. tuple pointers. In this manner we restrict the number of values that needs to be checked against a regular expression. What is critical in such an approach is the strategy for dropping elements from the cache when it is full, where we weigh how frequent a substring is in queries and how selective it is with the data, in order to retain the most valuable cache elements that both appear often in queries and drastically restrict the volume of values that needs to be checked against the regular expression.

3.2 Ontology Alignment

This Ontology Alignment Tool is a semi-automatic component that will employ / integrate different alignment methods in order to provide the vocabulary mappings needed for querying heterogeneous sources.

During the first year, we have reviewed the state of the art and carried out preliminary experiments, focusing on synthesis approaches and collaborative, semi-automatic alignment methods and the comparison of automatic alignment tools. A first version of SYNTHESIS, the automatic alignment platform, incorporating four individual alignment methods, was implemented and participated in the OAEI 2013 Campaign. The platform synthesizes the results of the underlying methods, dynamically allocating a subset based on the characteristics of a given alignment task and aiming at maximizing the social welfare of the interacting parties.

Furthermore, a first prototype of a GUI for human-assisted alignment was developed, and work on producing a standardized API for the semi-automatic environment has begun. Finally, we are carrying out an initial investigation of methods for improving on the scalability of semi-automatic alignment systems. Additionally, we have interacted with the Ontology-Lexica Community Group (Ontolex) of the W3C and contributed linguistic annotations-based alignment as a use case for Ontolex, as well as an API for editing and creating linguistic metadata in the LIME format.

4. 1st Year Investigation Activities

4.1 Content Classification & Ontology Evolution

Content Classification and Ontology Evolution Tools will incorporate (a) content classification methods for automatically annotating content with a finer schema than its current annotations, in situations where the granularity difference between two schemas makes them un-alignable; (b) ontology evolution methods for recommending refinements for a coarser schema, such that it will align better with schemas it is often queried in conjunction with.

During the first two months of the relevant task we have carried out an initial investigation of suitable methods.

4.2 Heterogeneous Distributed Semantic Querying

The Query Decomposition and the Federated Endpoint Wrapper components of the SemaGrow Stack will extend and adapt distributed querying methods and systems, so that they can exploit the results of *resource discovery* and implement a querying strategy that dispatches query fragments to those sources that are most likely to yield results. During the first two months of this task we have carried out an initial investigation of suitable methods.

5. Large-scale Experimentation

5.1 Semantic Store Infrastructure

The SemaGrow Stack will be deployed over a large-scale computational infrastructure that will be used for the project's experiments on distributed semantic stores.

During the first year of the project, the PARADOX cluster where the SemaGrow large-scale experiments will take place underwent a major update, leading in significant increases in computational power and storage capacity. Access to the infrastructure has been provided via various interfaces that were appropriately set up (batch, gLite, gUSE, RESTful interface).

5.2 Scalability & Robustness Experimental Methodology

The Scalability & Robustness Experimental Methodology will provide the guidelines for the development of automatic rigorous testing components that allow the project to reliably measure and compare the efficiency of the developed research components system under realistic conditions. The methodology will take into account individualities that occur due to distinct properties of the system such as the heterogeneity and the distributed nature of the repositories.

The methodology will define the measures for evaluating both the distinct components that realize the core SemaGrow research outcomes, as well as, the overall performance of the system. The following table summarizes the performance measures for the individual SemaGrow research outcomes.

SemaGrow Research Outcomes	Performance Measure
Novel indexing algorithms that support the efficient storage and retrieval of data summaries that concisely describe instance-level metadata about the different sources federated under the SemaGrow infrastructure.	Success will be measured in terms of (a) the size of the data summaries as a function of the total size of the federated repositories; (b) the overhead of the method as a function of the time it would take to query all repositories; and (c) the accuracy of the source selection in predicting which sources hold data that satisfy a given query.
An extension of state-of-the-art query decomposition and rewriting methods that will enable complex queries in one schema to be broken down into sub-queries, each in a (possibly) different schema.	Success will be measured in terms of selecting the optimal decomposition, achieving good load balancing between sources while at the same time minimizing rewriting, especially when the source schema is structurally distanced from the query schema.
The integration of a variety of state-of-the-art schema alignment methods under a novel architecture for the prior selection of the most appropriate method or methods for a given schema pair, the synthesis of multiple methods into a unified alignment, and the posterior evaluation of alignment quality.	Success will be measured in terms of (a) the performance of the alignment synthesis versus the performance of the best of the components being synthesised; and (b) the F-score of the tuples retrieved when compared with the tuples retrieved by hand-crafted queries.

Overall, the SemaGrow POWDER store will be evaluated against both current POWDER implementations and against state of-the-art non-POWDER stores where POWDER-inferred triples have been made explicit. With respect to the latter, we will test against the best large-scale stores that are freely available or for which academic research licenses can be obtained, such as Virtuoso, 4store, or bigdata. In this case, the evaluation will comprise four metrics: (a) compression in number of triples, (b) compression in disk volume, (c) responsiveness, measured as the time to retrieve the first query result and time between successive query results, and (d) throughput, measured as the time to retrieve all query results.

Finally, the *reactivity and scalability of the overall distributed system* will be measured in terms of (a) the size of the data summaries needed by the source selection algorithm as a function of the total size of the federated repositories; (b) the

overhead of the method as a function of the time it would take to query all repositories; and (c) the accuracy of the source selection in predicting which sources hold data that satisfy a given query.

The queries used for these measurements will be derived by combining elicitation from domain experts with analysis of the query logs of the currently deployed services as well as of the logs of the early SemaGrow deployments. As a result, the Scalability & Robustness Experimental Methodology should evolve to cover the needs and findings of the undergoing experiments, the methodology will be refined in sync with the progress of the SemaGrow components.

6. Real-life Experimentation

6.1 Envisaged Applications & Use Cases

The Use Cases foreseen in SemaGrow are classified under three categories, covering the different aspects of real-life experimentation:

- *Heterogeneous Data Collections & Streams*: The perspective from which extremely large and very complex agriculture-related data sets are considered is the one of research activities, during which the users need to cope with heterogeneous data collections & streams in order to achieve new scientific investigations that may help forecast and address societal challenges such as food production in changing climate conditions.
- *Reactive Data Analysis*: The perspective from which extremely large and very complex agriculture-related data sets are considered is the one of information management, during which the users need to cope with reactive analysis of the data within the time scale and processes that they need to support in order to create value through extensive data collection and analysis that may help timely and better decision making related to societal challenges like food security.
- *Reactive Resource Discovery*: The perspective from which extremely large and very complex agriculture-related data sets are considered is the one of education, during which the users need to cope with reactive resource discovery in order to be able to find, reuse and exploit data resources created in one environment in very different contexts.

6.2 Real-life Deployment & User Evaluation Plan

In order to realize an evaluation of the system in real-life situations, a detailed plan for the implementation of three (3) service demonstrators on top of the Semantic Store will be designed. The plan will describe how the demonstrators should be developed and deployed and include a methodology for evaluating user satisfaction.

7. Data Preparation

The content providing members of the consortium (DLO, FAO, AK) have direct ownership or access to all the data sources that will be used within SemaGrow. These data sources already exist and are maintained independently of SemaGrow, are available to SemaGrow from day one of the project, and will continue to be available and maintained beyond project's end.

These data sources are expressed using various and different formats (e.g. XML, JSO, NetCDF, etc.). For the purposes of SemaGrow, this data must undergo a triplification process in order to be introduced in the semantic infrastructure.

To this end, the appropriate triplification modules were designed and implemented in order to transfer the initial non-triplified data into triples.

The following picture depicts the technological components involved in the triplification process for SemaGrow collections that are expressed in different formalizations.

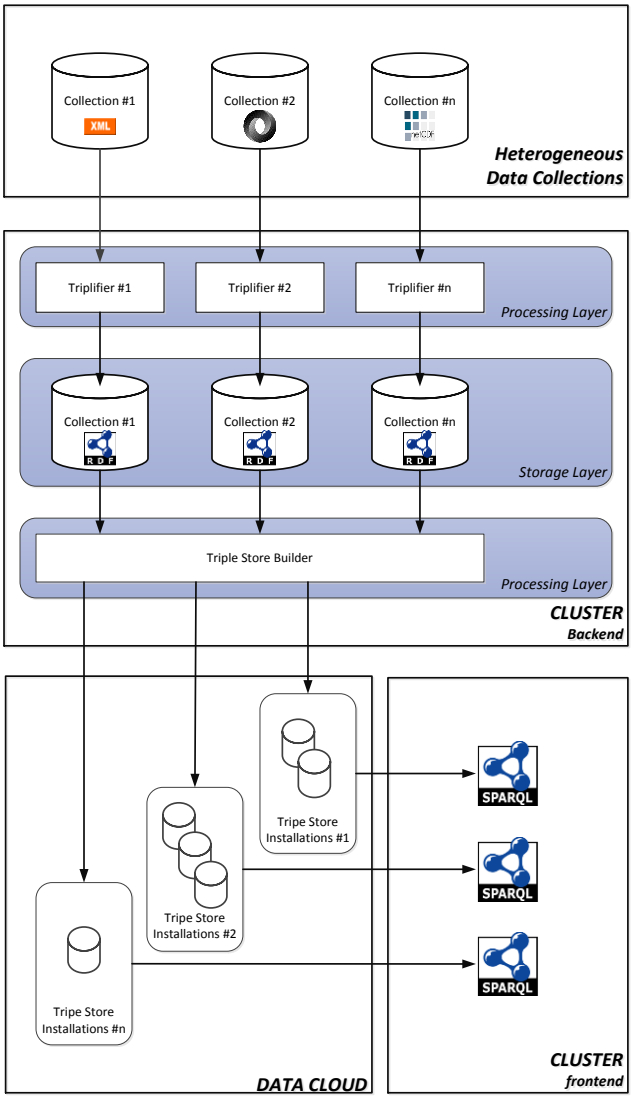

Figure 2: Triplification Architecture

The first step towards triplifying data collections is to store them in their original format (e.g. XML files, NetCDF files, JSON snippets etc.). For each collection, a triplifier module is activated in order to produce the corresponding RDF/XML file for each original entry. At a first level, the RDF/XML outputs are stored as files in the file system. Following this, the Triple Store Builder is the component responsible for ingesting the produced triples into Triple Store installations. Depending on its size, a collection may require multiple triple store instances for storing the triples included in the

collection. The Triple Stored Builder will be responsible for creating new Triple Store instances when necessary, directing the insertions into the appropriate triple store instance and load balancing the insertion process. Finally, the complete set of Triple Store instantiations for each collection is exposed via a single SPARQL interface.