A Data Lake Metadata Enrichment Mechanism via Semantic Blueprints

Michalis Pingos and Andreas S. Andreou
Department of Computer Engineering and Informatics, Cyprus University of Technology, Limassol, Cyprus

Keywords: Smart Data Processing, Deep Insight, Data Lakes, Heterogeneous Data Sources, Metadata Mechanism, 5Vs Big Data Characteristics, Data Blueprints.

Abstract: One of the greatest challenges in Smart Big Data Processing nowadays revolves around handling multiple heterogeneous data sources that produce massive amounts of structured, semi-structured and unstructured data through Data Lakes. The latter requires a disciplined approach to collect, store and retrieve/ analyse data to enable efficient predictive and prescriptive modelling, as well as the development of other advanced analytics applications on top of it. The present paper addresses this highly complex problem and proposes a novel standardization framework that combines mainly the 5Vs Big Data characteristics, blueprint ontologies and Data Lakes with ponds architecture, to offer a metadata semantic enrichment mechanism that enables fast storing to and efficient retrieval from a Data Lake. The proposed mechanism is compared qualitatively against existing metadata systems using a set of functional characteristics or properties, with the results indicating that it is indeed a promising approach.

1 INTRODUCTION

Big Data is an umbrella term referring to the large amounts of digital data continually generated by tools and machines, and the global population (Chen et al., 2014). The speed and frequency by which digital data is produced and collected by an increasing number of different kinds of sources are projected to increase exponentially. This increasing volume of data, along with its immense social and economic value (Bertino, 2013; Günther et al., 2017), is driving a global data revolution. Big Data has been called “the new oil”, as it is recognized as a valuable human asset, which, with the proper collation and analysis, can deliver information that will give birth to deep insights into many aspects of our everyday life and, moreover, to let us predict what might happen in the future.

While Big Data is available and easily accessible, it is evident that its great majority comes from heterogeneous sources with irregular structures (Blazquez & Domenech, 2018). The process of transforming Big Data into Smart Data in terms of making them valuable and transforming it into meaningful information is called Smart Data Processing (SDP) and includes a series of diverse actions and techniques. These actions and techniques support the processing and integration of data into a unified view from disparate Big Data sources. Furthermore, this field includes adaptive frameworks and tool-suites to support smart data processing by allowing the best use of streaming or static data, and may rely on advanced techniques for efficient resource management.

SDP supports the process and integration of data into a unified view from disparate Big Data sources including Hadoop and NoSQL, Data Lakes, data warehouses, sensors and devices on the Internet of Things, social platforms, and databases, whether on-premises or on the Cloud, structured or unstructured and software as a service application to support Big Data analytics (Fang, 2015).

The analytics solutions, which rely on smart data processing and integration techniques, are called Systems of Deep Insight (SDI). These solutions enable optimization of asset performance in SDP systems and are geared towards systems of insight. In addition, they sift through the data to discover new relationships and patterns by analysing historical data, assessing the current situation, applying business rules, predicting outcomes, and proposing the next best action. Despite the great and drastic solutions proposed in recent years in the area of Big Data Processing and SDP, treating Big Data produced

Pingos, M. and Andreou, A.
A Data Lake Metadata Enrichment Mechanism via Semantic Blueprints.
DOI: 10.5220/0011080400003176
In *Proceedings of the 17th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2022)*, pages 186-196
ISBN: 978-989-758-568-5; ISSN: 2184-4895

by multiple heterogeneous data sources remains a challenging and unsolved problem.

The main research contributions of this paper include the utilization of Data Lakes as a means to achieve the desired level of Big Data Processing and ultimately lead to developing SDP. Along these lines, we propose a standardization framework for storing data (and data sources) in a Data Lake and a metadata semantic enrichment mechanism that is able to handle effectively and efficiently Big Data coming from disparate and heterogeneous data sources. These sources produce different types of data at various frequencies and the mechanism is applied both when this data is ingested in a Data Lake and at the extraction of knowledge and information from the Data Lake.

The remainder of the paper is structured as follows: Section 2 discusses related work and the technical background in the areas of SDP and Data Lakes. Section 3 presents the Data Lake source description framework and discusses its main components. This is followed by presenting the details of the expected features of a Data Lake's metadata system and compares them with several existing works in section 4. Finally, section 5 concludes the paper and highlights future work directions.

2 TECHNICAL BACKGROUND

Big Data Processing includes the processing of multiple and various types of data, structured, semi-structured, and unstructured. A Data Lake is a storage repository that could store a vast amount of raw data of these various types in its native format until it is needed. In addition, a Data Lake is a centralized repository that stores structured, semi-structured, and unstructured data at any scale, and data is selected and organized when needed. Data can be stored as-is, without the need to first structure it before executing different types of analytics, from dashboards and visualizations to Big Data processing, real-time analytics, and machine learning to guide better decisions. Furthermore, Data Lakes architecture is also used to store large amounts of relational and non-relational data combining them with traditional data warehouses.

Khine and Wang (2018) state that a Data Lake is one of the arguable concepts that appeared in the era of Big Data. A Data Lake is a place to store practically every type of data in its native format with no fixed limits on account size or file, offering at the same time high data quantity to increase analytic performance and native integration.

The idea behind Data Lakes is simple: Instead of placing data in a purpose-built data store, move it into a Data Lake in its original format. This eliminates the upfront costs of data ingestion, like transformation and indexing. Once data is placed into the lake, it is available for analysis by everyone in the organization (Miloslavskaya & Tolstoy, 2016). Unlike a hierarchical Data Warehouse where data is stored in files and folders, a Data Lake has a flat architecture. Every data element in a Data Lake is given a unique identifier and is tagged with a set of metadata information. Data Lakes can provide the following benefits:

- With the onset of storage engines like Hadoop, storing disparate information has become easy.
- There is no need to model data into an enterprise-wide schema with a Data Lake.
- With the increase in data volume, data quality, and metadata, the quality of analyses also increases.
- Data Lake offers business Agility.
- Machine Learning and Artificial Intelligence can be used to make profitable predictions.
- It offers a competitive advantage to the implementing organization.

The 5Vs are the five main and innate characteristics of Big Data. As one-dimension changes, the likelihood increases that another dimension will also change as a result. Knowing the 5Vs allows data scientists to derive more value from their data, while also allowing them to become more customer-centric (Bell et al., 2021). As far back as 2001, Doug Laney articulated the now mainstream definition of Big Data as the 3Vs of Big Data (Kościelniak & Putto, 2015):

- **Volume** - the volumes of data produced
- **Velocity** - the rate transfer which data produced
- **Variety** - heterogeneous and multiple data

In addition to the 3Vs, other dimensions of Big Data have also recently been reported (Gandomi & Haider, 2015). These include:

- **Veracity:** IBM coined Veracity as the 4th V, which represents the unreliability inherent in some sources of data (Luckow et al., 2015).
- **Variability:** SAS introduced Variability and Complexity as two additional dimensions of Big Data (Herschel & Miori, 2017). Variability refers to the variation in the data flow rates.

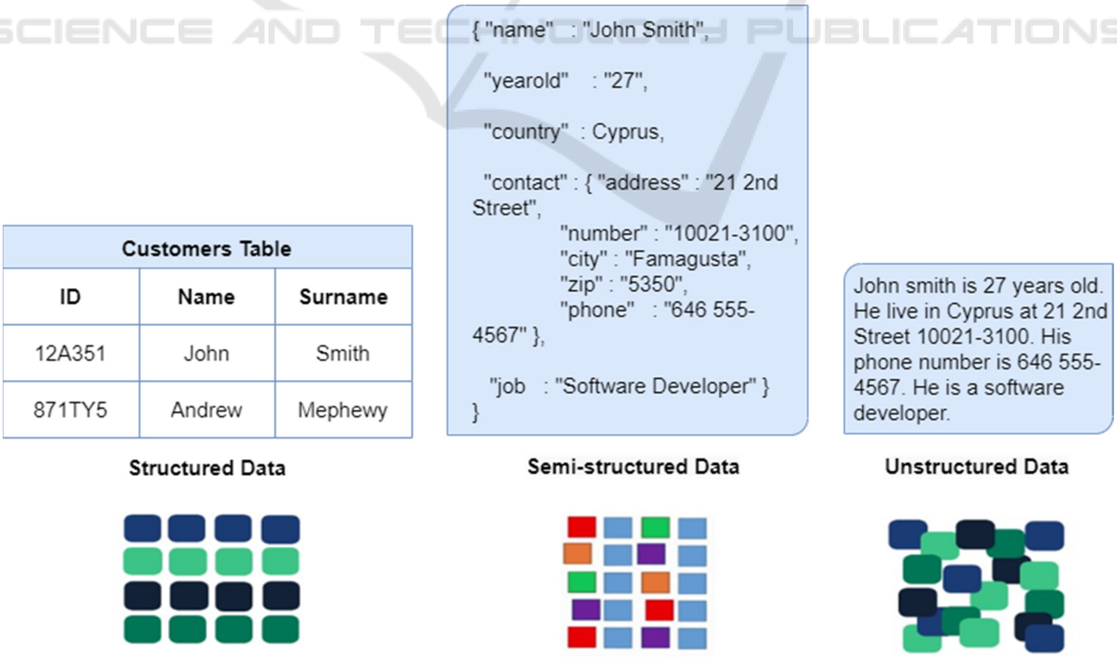
Complexity refers to the fact that Big Data are generated through a myriad of sources.

- **Value:** Oracle introduced Value as a defining attribute of Big Data (Kim et al., 2012). Based on Oracle's definition, Big Data are often characterized by relatively "low value density".

Big Data originate mostly from one of five primary sources: Social media, Cloud, Web, traditional business systems (e.g., ERPs), and Internet of Things (IoT) (Sethi & Sarangi, 2017). These primary sources produce an enormous variety of structured, unstructured, and semi-structured data (see figure 1). The term structured data refers to data that resides in a fixed field within a file or record. Structured data is typically stored in a relational database (RDBMS). It depends on the creation of a data model that defines what types of data are present. Unstructured data is more or less all the data that is not structured. Even though unstructured data may have a native, internal structure, it is not structured in a predefined way. There is no data model; the data is stored in its native format. Typical examples of unstructured data are rich media, text, social media activity, surveillance imagery, etc. It is essentially a type of structured data that does not fit into the formal structure of a relational database. But while not matching the description of structured data entirely, it still employs tagging systems or other markers, separating different elements and enabling search. Sometimes, this is referred to as data with a self-describing structure.

Media includes social networks and interactive platforms, like Google, Facebook, Twitter, YouTube, Instagram, as well as generic media, like videos, audios, and images that provide qualitative and quantitative data on every aspect of user interaction. Private or public cloud storages include information from real-time or on-demand business data. Web or Internet constitutes any type of data that are publicly available and can be used for any commercial or individual activity. Traditional business systems produce and store business data in conventional relational databases or modern NoSQL databases. Finally, IoT includes data generated from sensors that are connected to any electronic devices that can emit data.

Manufacturing blueprints create a basic knowledge environment that provides manufacturers with more granular, fine-grained and composable knowledge structures and approaches to correlate and systematize vast amounts of dispersed manufacturing data, associate the "normalized" data with operations, and orchestrate processes in a more closed-loop performance system that delivers continuous innovation and insight. Such knowledge is crucial for creating manufacturing smartness in a smart manufacturing network (Papazoglou & Elgammal, 2018).



Customers Table

ID	Name	Surname
12A351	John	Smith
871TY5	Andrew	Mephewy

```
{ "name"   : "John Smith",
  "yearold"   : "27",
  "country"  : Cyprus,
  "contact" : { "address" : "21 2nd
Street",
              "number" : "10021-3100",
              "city" : "Famagusta",
              "zip" : "5350",
              "phone"   : "646 555-
4567" },
   "job   : "Software Developer" }
}
```

John smith is 27 years old. He live in Cyprus at 21 2nd Street 10021-3100. His phone number is 646 555-4567. He is a software developer.

Figure 1: Different types of data produced by Big Data sources.

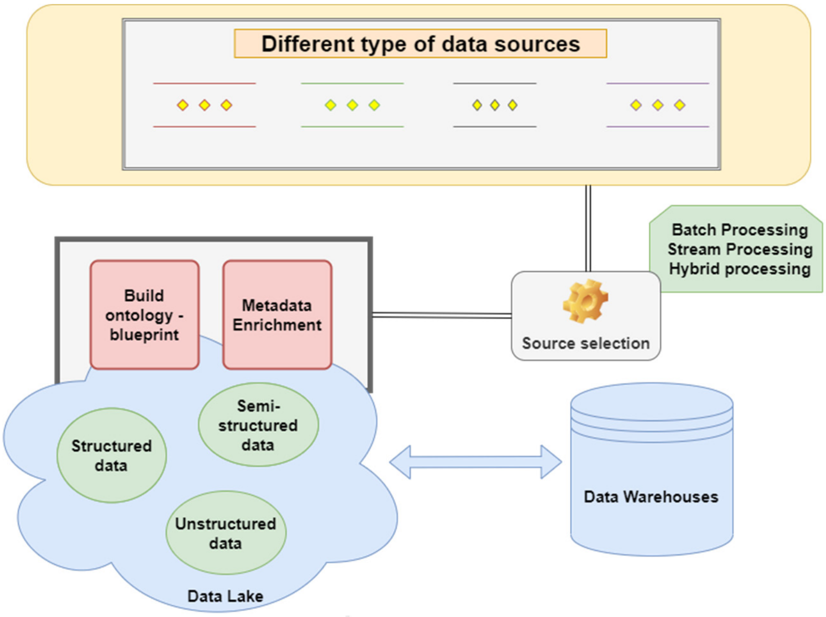


Figure 2: Data sources selection metadata enrichment mechanism using 5Vs.

Manufacturing blueprints provide a complete summary of a product by juxtaposing its features with its operational and performance characteristics, as well as how it is manufactured, which processes are used, and which manufacturing assets (people, plant machinery and facilities, production line equipment) are used to make it, as well as the suppliers who provide/produce parts and materials. Manufacturing blueprints also include a summary of suppliers, including their capabilities and competencies. Finally, manufacturing blueprints detail how manufacturers and suppliers coordinate, arrange manufacturing processes, expedite hand-offs, and create the final product (Papazoglou & Elgammal, 2018).

This paper adopts the basic principles of manufacturing blueprints and modifies their purpose and meaning to reflect the description and characterization of data sources and the data they produce. Along these lines, a framework is proposed that builds upon the utilization of the five aforementioned Big Data characteristics (5Vs) to describe Big Data sources. These characteristics will guide the characterization of data sources by means of specific types of blueprints through an ontology-based description. Big Data sources will thus be accompanied by this blueprint description before they become part of a Data Lake.

3 METHODOLOGY

As mentioned above, a new standardization framework will be introduced combining the 5Vs Big Data characteristics and blueprint ontologies to assist data processing (storing and retrieval) in Data Lakes organized with pond architecture. According to the pond architecture, a Data Lake consists of a set of data ponds and each pond hosts / refers to a specific data type. Each pond contains a specialized storage system and data processing depending on the data type (Sawadogo & Darmont, 2021).

A Data Lake with pond architecture is assumed to use a dedicated pond to store each data source with the same type of data, structured, unstructured, and semi-structured as shown in figure 2. This innate pond architecture is particularly helpful when extracting information from the Data Lake as will be demonstrated later on.

Big Data sources are filtered before they become part of the Data Lake as shown in figure 2. Every data source, which is a candidate to be part of the Data Lake, will be characterized according to the blueprint values shown in figure 3. Therefore, the selection of data sources is performed according to the blueprint of each different data source.

As previously mentioned, a dedicated blueprint is developed to describe each data source storing data in the Data Lake. Specifically, the blueprint of a source consists of two interconnected blueprints as shown in figure 3:

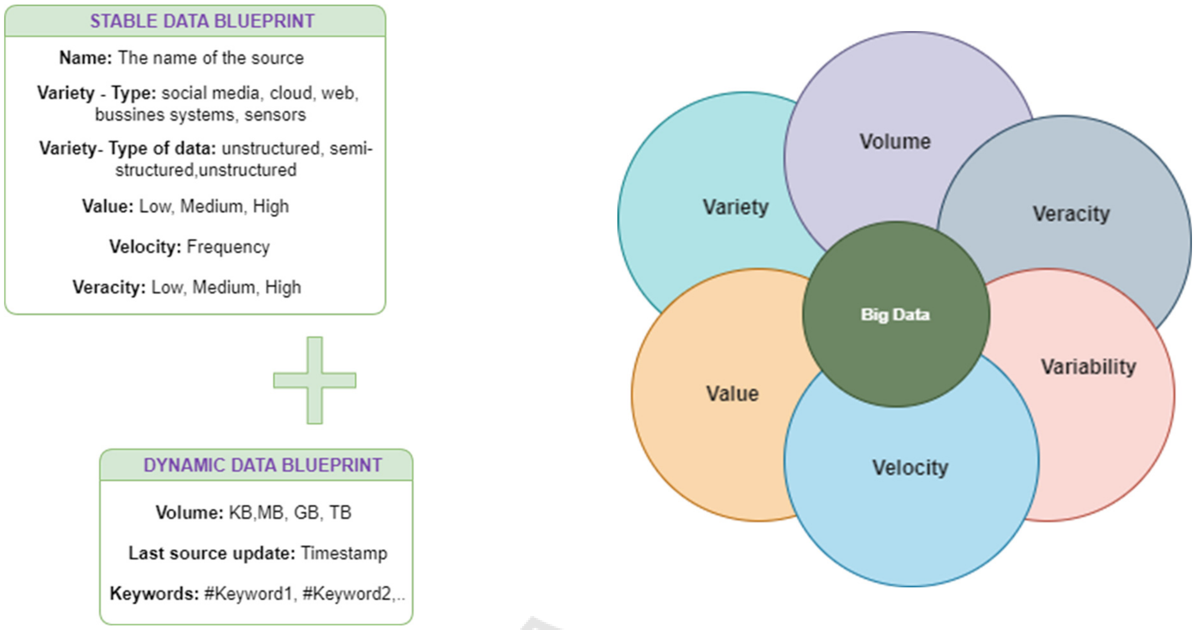

STABLE DATA BLUEPRINT
Name: The name of the source
Variety - Type: social media, cloud, web, bussines systems, sensors
Variety- Type of data: unstructured, semi-structured,unstructured
Value: Low, Medium, High
Velocity: Frequency
Veracity: Low, Medium, High
DYNAMIC DATA BLUEPRINT
Volume: KB,MB, GB, TB
Last source update: Timestamp
Keywords: #Keyword1, #Keyword2,..
Volume
Variety
Veracity
Big Data
Value
Variability
Velocity

Figure 3: Data source blueprints description using 5Vs Big Data characteristics.

The first one is static and records the name and type of the source, the type of data it produces, as well as the value, velocity, variety, and veracity of the data source. The second is dynamic as it changes values in the course of time and essentially characterizes the volume of data, the last source update, and the keywords of the source. The dynamic blueprint is updated every time data sources produce new data. Figure 5 presents the ontology graph of the data source created via Protégé, a free open-source ontology editor and framework for building knowledge-based solutions (http://protege.stanford.edu). This graphical representation tool produces an RDF ontology for the data sources.

RDF stands for Resource Description Framework and is a framework for describing resources usually on the Web. RDF is designed to be read and understood by computers, is not designed for being displayed to people, and is written in XML. RDF is a part of the W3C's Semantic Web Activity and is a standard for data interchange that is used for representing highly interconnected data. Each RDF statement is a three-part structure consisting of resources where every resource is identified by a URI. Representing data in RDF allows information to be easily identified, disambiguated and interconnected by AI systems.

We use the Resource Description Framework to describe a source's stable and dynamic blueprint with the combination of the theory of triples (subject, predicate, object) (see figure 4). For example, let us assume that we wish to store values in a Data Lake produced by three candidate sources and that these sources bear the following characteristics – attributes according to the data source blueprints (see figure 3):

- Source 1

  Stable Blueprint Attributes:

  Name: Source 1
  Variety-Type: Sensor
  Variety-Type of data: Unstructured
  Value: High
  Velocity: 1sec
  Veracity: Medium

  Dynamic Blueprint Attributes:

  Volume: KB
  Last Update: 24/01/2022; 08:34
  Keywords: # Products delivery

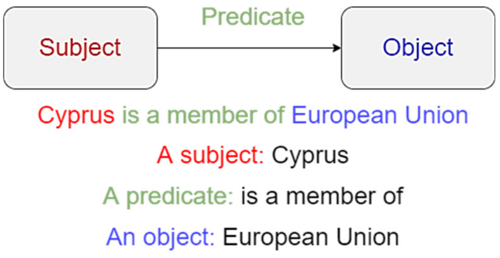

Predicate
Subject
Object
Cyprus is a member of European Union
A subject: Cyprus
A predicate: is a member of
An object: European Union

Figure 4: The basic semantic RDF triple model.

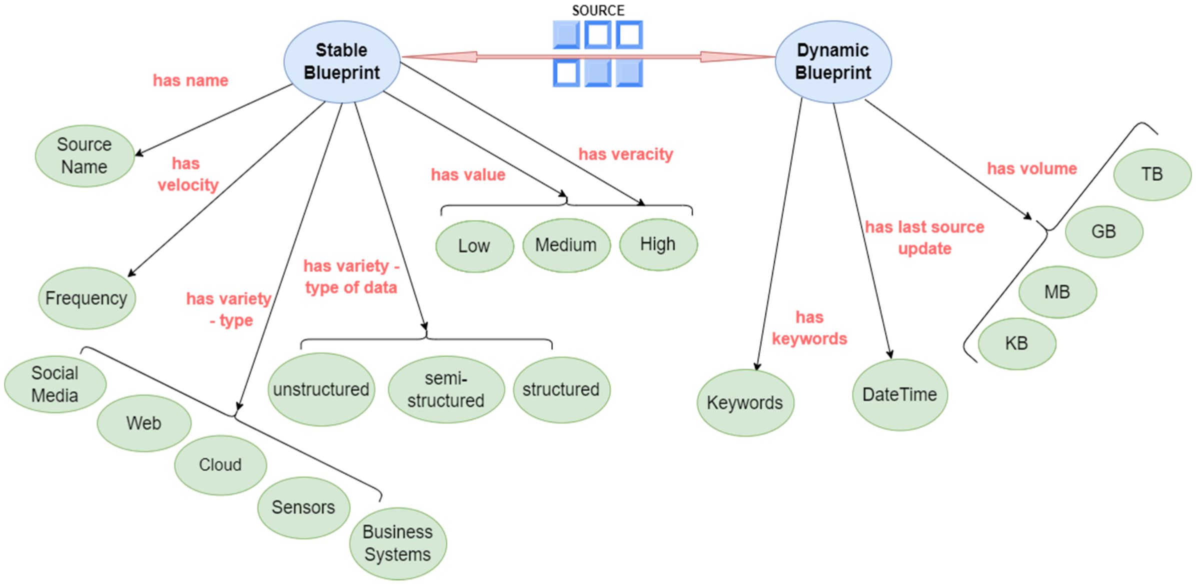


Figure 5: Stable and Dynamic data source blueprint ontology graph.

- Source 2
  - *Stable Blueprint attributes*:
    - **Name**: Source 2
    - **Variety-Type**: Business Systems
    - **Variety-Type of data:** structured
    - **Value:** High
    - **Velocity:** 1sec
    - **Veracity:** Medium
  - *Dynamic Blueprint attributes*:
    - **Volume:** KB
    - **Last Update**: 24/01/2022 08:34
    - **Keywords:** # Products delivery

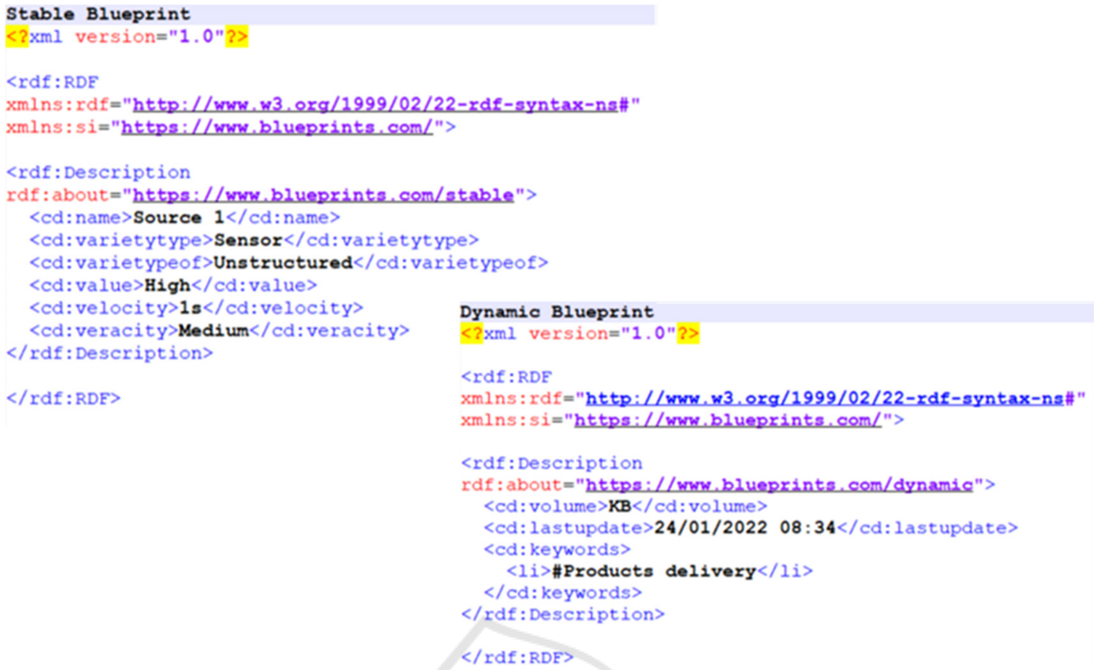
- Source 3
  - *Stable Blueprint attributes*:
    - **Name**: Source 3
    - **Variety-Type**: Web
    - **Variety-Type of data:** unstructured
    - **Value:** High
    - **Velocity:** 12Hours
    - **Veracity:** High
  - *Dynamic Blueprint attributes*:
    - **Volume:** KB
    - **Last Update**: 24/01/2022 06:50

By using SPARQL (Protocol and Query Language), or other methods, we may query all RDF resources before being ingested into the Data Lake or after their ingestion. This requires that each data source has its description in RDF form as mentioned before, which can be retrieved by a public API (RDF API) for the sources to be queried. Figure 6 shows the RDF files written in XML for Source 1 of the given example based on the blueprint ontology description created. The XML follows the same structure for every source according to its characteristics.

Let us now assume that all three sources described above will become members of our Data Lake. Therefore, we must first build a specific part of the Data Lake that consists of data sources with Value - High, Veracity – Medium OR High, AND Keywords - # Products delivery. A source selection middleware is fed with these preferred rules (see figure 1) and executes the following SPARQL query:

```
SELECT ? sources
   WHERE {
         ? source <has value>  High  &&
          <has veracity>  Medium  &&
      <has keyword>  #Products delivery
      }
```

Once this selection process is completed, the RDF created earlier becomes now part of the Data Lake and contributes to the Data Lake's metadata semantic enrichment which is the cornerstone for addressing the challenge of easy storing and efficient retrieval of data. The result of the query consists of the stable and dynamic blueprints of Source 1 and Source 2, which satisfy the query parameters and thus will be added to the Data Lake's RDF schema.



```
Stable Blueprint
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:si="https://www.blueprints.com/">

<rdf:Description
rdf:about="https://www.blueprints.com/stable">
  <cd:name>Source 1</cd:name>
  <cd:varietytype>Sensor</cd:varietytype>
  <cd:varietypeof>Unstructured</cd:varietypeof>
  <cd:value>High</cd:value>
  <cd:velocity>1s</cd:velocity>
  <cd:veracity>Medium</cd:veracity>
</rdf:Description>

</rdf:RDF>
```

```
Dynamic Blueprint
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:si="https://www.blueprints.com/">

<rdf:Description
rdf:about="https://www.blueprints.com/dynamic">
  <cd:volume>KB</cd:volume>
  <cd:lastupdate>24/01/2022 08:34</cd:lastupdate>
  <cd:keywords>
    <li>#Products delivery</li>
  </cd:keywords>
</rdf:Description>

</rdf:RDF>
```

Figure 6: Stable and Dynamic Blueprint for Source 1.

The selected data sources are then distributed to the specific Data Lake Pond for further processing according to the corresponding attribute values. Essentially, this process and the associated characterization help to handle and manage multiple and diverse types of data sources and to contribute to the Data Lake's metadata enrichment before and after these sources become members.

When a data source becomes part of the Data Lake, from that point forward the metadata schema is utilized for filtering and retrieving data based on the blueprints and their metadata. The latter involves attributes such as the type of data produced by the sources, the size of the data they produce, the speed of production, the accuracy of the data, and the importance of the source data, etc. Therefore, each action for retrieving data from the Data Lake is effectively guided by the information provided in the metadata mechanism, that is, in the blueprints.

Especially in the case of the dynamic blueprint, this portion of the metadata will dynamically be updated each time new data is produced by the sources, or when deemed necessary (e.g., when changing the location of the associated pond).

To further demonstrate the applicability and the value that the proposed metadata mechanism brings to supporting the data actions in a Data Lake, and in particular the retrieval of data, let us use once again the example of the sources given earlier. As previously mentioned, the selected data sources are distributed to the specific Data Lake pond for further processing according to the corresponding attribute values. After the completion of the selection process, the retrieval process is based on the metadata semantic enrichment – RDF schema of the Data Lake encoded in the blueprints. Let us now assume Source 1 is a member of the pond with structured data and that Source 2 is also a member of the pond with unstructured data. If we wish to retrieve all the product delivery data from our Data Lake by a middleware residing between the Data Lake and the application layer of a system that uses the Data Lake, then in the simplest case all that needs to be done is to execute the following SPARQL query:

```
SELECT ? Dlsources
 WHERE {
     ? source <has keyword> #Products delivery
       }
```

Essentially, this performs a classic retrieval action from the Data Lake and the result is to push all the data sources with the Product Delivery keyword to the application layer. Therefore, the data retrieved are larger in volume and the complexity to filter them after retrieval is higher. If we utilize the metadata information offered by the semantic enrichment mechanism, then we can refine the type of information sought in the Data Lake and get the results we need focusing on specific values or attributes. For example, using the attributes shown in figure 5 it is feasible to execute more guided queries such as:

```
SELECT ? DLsources
 WHERE {
    ? source <has value> High &&
    <has Variety-Type of data> Unstructured &&
     <has keyword> #Product delivery
    }
```

These guided queries can range from simple to more sophisticated by utilizing the full spectrum of the 5Vs data characteristics mentioned in figure 3. Thus, they allow data scientists to derive more value from their data and to define custom levels of granularity and refined information in the data sought as required. Essentially, this SPARQL query process and the associated characterization support the handling and management of multiple and diverse types of data sources residing in a Data Lake in a simple yet efficient way.

4 PRELIMINARY VALIDATION

Sawadogo et al. (2019) identified six main functional characteristics that should ideally be provided by a Data Lake metadata system:

- Semantic Enrichment (SE)
- Data Indexing (DI)
- Link generation and conservation (LG)
- Data Polymorphism (DP)
- Data Versioning (DV)
- Usage Tracking (UT)

Semantic Enrichment consists in generating a description of the context of data (e.g., tags) using knowledge bases such as ontologies. *Semantic Enrichment* summarizes the datasets contained in the lake to make it understandable and to identify data links. For instance, data associated with the same tags can be considered linked. Our mechanism meets this characteristic since it utilizes both the dynamic and the stable blueprint based on the basic RDF triple model presented in figures 4 and 5. The second main functionality identified by Sawadogo et al. (2019) is *Data Indexing* which includes setting up a data structure to retrieve datasets based on specific keywords or patterns. This functionality provides optimization of data querying in the Data Lake through keywords filtering. This characteristic is offered in our metadata semantic enrichment mechanism via the attribute Variety - type of data, which is used to distribute data sources and data ponds according to their structure and the keyword attribute in the stable blueprint as presented in figure 3. *Link generation and conservation* is the process of detecting similarity relationships or integrating pre-existing links between datasets to identify data clusters, data groups where data are strongly linked to each other and significantly different from other data. Our mechanism provides this functionality via the keyword attributes in the dynamic blueprint which is updated every time a new data source or new data that is produced by a registered source are pushed to the Data Lake. *Data Polymorphism* is defined as storing multiple representations of the same data and *Data Versioning* refers to the ability of the metadata system to support data changes during the processing in the Data Lake. These functional characteristics are provided by our metadata semantic enrichment mechanism via the process of storing the metadata description - blueprint every time sources in the Data Lake change or produce new data. When new data is pushed into the Data Lake a new timestamped representation of this data is created and stored in the Data Lake along with the existing representations-blueprints. During the data processing in the Data Lake, the proposed mechanism updates the Dynamic Blueprint, especially the keywords if deemed necessary. Finally, the mechanism provides also the last referred functionality *Usage Tracking* which is the process of recording the interactions between users and the Data Lake. Essentially, when data is queried a timestamp accompanied by the user details that executed the last query are recorded in the Dynamic Blueprint.

Additionally, Sawadogo et al. (2019) provide a synthetic comparison of 15 metadata systems. We selected the two most completed systems examined in that paper in terms of functionality, that is, CoreKG (Beheshti et al., 2018) and MEDAL (Sawadogo et al., 2019), with the aim to compare them with our metadata data mechanism using a set of new functional characteristics introduced in this paper. These characteristics can add value to the synthetic examination of the quality and efficiency of metadata enrichment mechanisms for Data Lakes. The new characteristics are:

- Granularity
- Ease of storing/retrieval
- Size and type of metadata
- Expandability

We define *Granularity* as the ability to refine the type of information that needs to be retrieved using for example keywords. This ability is expressed by the number of fine-grained levels the metadata mechanism supports for defining the information sought. *Ease of storing/retrieval* refers to the ability of the metadata mechanism to store or retrieve data in the Data Lake in a simple and easy way. It is assumed here that the retrieval action is efficient enough to return the desired parts of the information sought. This characteristic is reflected on the number of steps that need to be executed for the process of storing and retrieving data items to be completed. The *Size and type of metadata* refers to the volume and the kind of metadata that are produced by the mechanism and which are necessary for the efficient and accurate retrieval of data. The larger the size and/or the higher the complexity of the type of data needed the lower the performance and suitability of the metadata mechanism. Finally, we define *Expandability* as the ability to expand the metadata mechanism with further functional characteristics or other supporting techniques and approaches, such as visual querying. Obviously, the more open the mechanism for expansion the better. These characteristics are evaluated using a Likert Linguistic scale, including the values Low, Medium, and High. Table 1 provides a definition of Low, Medium and High for each characteristic introduced.

Table 1: Definition of Low, Medium, and High of each characteristic.

Characteristic	**Low**	**Medium**	**High**
Granularity	1 level	2 levels	3 or more levels
Ease of storing/retrieval	5 or more actions	3-4 actions	2 actions maximum
Size of metadata	KB	MB	GB
Expandability	No or limited	Normal	Unlimited

As previously mentioned, we used the suggested characteristics to provide a short comparison between our metadata enrichment mechanism and the two top existing metadata mechanisms suggested by Sawadogo et al. (2019), that is, MEDAL and CoreKG.

MEDAL adopts a graph-based organization based on the notion of object and a typology of metadata in three categories: intra-object, inter-object, and global metadata. A hypernode represents an object containing nodes that correspond to the versions and representations of an object. MEDAL is modeled also by oriented edges which link the nodes providing transformation and update operations. Hypernodes of the mechanism can be linked in several ways, such as edges to model similarity links and hyperarcs to translate parenthood relationships and object groupings. Finally, global resources are present in the form of knowledge bases, indexes, or event logs. This concept and operation of the framework provide SE, DI, LG, DP, DV, and UT. As a result, MEDAL can be characterized with High *Granularity*, Medium *Ease of storing/retrieval* using indexes and event logs, Medium *Size and type of metadata*, and an undefined *Expandability* since no reference is made on how it can be evolved in the future.

CoreKG is an open-source complete Data and Knowledge Lake Service which offers researchers and developers a single REST API to organize, curate, index and query their data and metadata over time. At the Data Lake layer, CoreKG powers multiple relational and NoSQL database-as-a-service for developing data-driven Web applications. This enables the creation of relational and/or NoSQL datasets within the Data Lake, create, read, update, and delete entities in those datasets, and apply federated search on top of various islands of data. It also provides a built-in design to enable top database security threats (Authentication, Access Control and Data Encryption), along with Tracing and Provenance support. On top of the Data Lake layer, CoreKG curates the raw data in the Data Lake and prepares them for analytics. This layer includes functions such as extraction, summarization, enrichment, linking and classification. Another part of the mechanism is the Notion of Knowledge Lake, a centralized repository containing virtually inexhaustible amounts of both raw data and contextualized data as a result to providing the foundation for Big Data analytics by automatically curating the raw data in data islands so as to provide insights from the vastly growing amounts of local, external and open data. This open-source service provides SE, DI, LG, DP, and UT. Based on the proposed properties scheme, CoreKG can be evaluated to have High *Granularity*, Medium *Ease of storing/retrieval* using the single API, with Medium *Size and type of metadata*, and High *Expandability* due to the use of the Hadoop ecosystem.

As described in section 4, the proposed metadata enrichment mechanism provides DI, LG, DP, DV and UT. Furthermore, our mechanism presents High *Granularity*, High *Ease of storing/retrieval* using the

stable and dynamic data source blueprint descriptions, with a Medium *Size and type of metadata*, and High *Expandability*. These values are attributed as follows:
High *Granularity* is achieved by the use of keywords that describe the sources and the blueprint values. This enables the user to define details at the level of the properties offered by these keywords and the type of blueprint characteristics for which values are kept. This list of features may be considered quite rich to enable the retrieval of data based on fine-grained query-like information. The High *Ease of storing/retrieval* is achieved by the blueprint description of the Data Lake as each time data sources are pushed to the Data Lake a variety of types of data attributes is produced, which helps the mechanism place the sources to a specific pond according to the structure of the data involved (structured, semi-structured and unstructured). This source distribution in the Data Lake facilitates simple and easy storing and retrieval of the information stored. Our framework is characterized by a low number of actions to: (1) Select and query data sources according to their stable and dynamic blueprint; (2) Push data in specific Data Lake Ponds. The *Size and type of metadata* produced by the mechanism has the maximum value of High due to the creation of the metadata description of the Data Lake every time new sources or data are pushed to the Data Lake and the DV characteristic that our blueprint provides by using the 5Vs Big Data characteristics. This may be considered a small overhead as it introduces a considerable number of metadata features, but their complexity is very low and their interpretation according to the 5Vs quite straightforward. Finally, our Data Lake implementation is based on the Hadoop ecosystem and hence this provides High *Expandability*. It should be noted here that expandability of the proposed mechanism can be traced in two aspects: (i) By using this simple semantic enrichment and blueprint ontologies we can easily apply visual querying during the source selection or data extraction; (ii) We may improve Data Lakes' privacy, security, and data governance, and, therefore, address some of the main challenges met with Data Lakes, by storing the descriptive metadata information to the blockchain. This enables storing of encrypted metadata information and may guarantee immutability of the metadata. Both aspects are currently under development as proof of concept, with very encouraging preliminary results thus far.

Table 2 sums up all the information of the short comparison between the three mechanisms made in this section. It is clear that the proposed mechanism seems to perform at least equally well, while in some characteristics it seems to prevail, such as *Ease of storing/retrieval* and *Expandability*.

Table 2: Evaluation and comparison of the mechanisms.

Characteristic	**MEDAL**	**CoreKG**	**Proposed approach**
Granularity	High	High	High
Ease of storing/retrieval	Medium	Medium	High
Size of metadata	Medium	Medium	Medium
Expandability	Undefined	High	High

5 CONCLUSIONS AND FUTURE WORK

This paper proposed a novel framework for standardizing the processes of storing/retrieving data generated by heterogeneous sources to/from a Data Lake organized with ponds architecture. The framework is based on a metadata semantic enrichment mechanism which uses the notion of blueprints to produce and organize meta-information related to each source that produces data to be hosted in a Data Lake. In this context, each data source is described via two types of blueprints which essentially utilize the 5Vs Big Data characteristics Volume, Velocity, Variety, Veracity and Value: The first includes information that is stable over time, such as, the name of the source and its velocity of data production. The second involves descriptors that vary as data is produced by the source in the course of time, such as the volume and date/time of production.

Every time data sources or data are pushed in and out of the Data Lake, the stable and dynamic blueprints are updated thus keeping a sort of history of these transactions. Essentially the description of the sources helps to treat and manage many, multiple, and different types of data sources and to contribute to Data Lakes' metadata enrichment before and after these sources become members of a Data Lake. When a data source becomes part of the Data Lake the metadata schema is utilized, describing the whole Data Lake ontology. The filtering and retrieval of data is based on this metadata mechanism which involves attributes from the 5Vs, attributes such as last source updates and keywords.

A short comparison to other existing metadata systems revealed the high potential of our approach as it offers a more complete characterization of the data sources and covers a set of key features reported in literature and expanded in this work. Furthermore, it provides the means to perform efficient and fast retrieval of the required information.

Future research steps will include the full implementation of the proposed mechanism using our metadata model in the context of structured, semi-structured and unstructured data. This will allow us to evaluate our framework in more detail, and in particular to compare it further against other existing systems via the use of certain performance metrics. This, in turn, will allow us to focus on and improve privacy, security, and data governance in Data Lakes by using the blockchain technology and smart contracts.

ACKNOWLEDGEMENTS

This paper is part of the outcomes of the CSA Twinning project DESTINI. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 945357.

REFERENCES

Chen, Min, Shiwen Mao, & Yunhao Liu. (2014). “Big Data: A Survey.” Mobile Networks and Applications 19(2): 171–209.

Bertino, Elisa. (2013). “Big Data - Opportunities and Challenges: Panel Position Paper.” Proceedings – International Computer Software and Applications Conference: 479–80.

Günther, Wendy Arianne, Mohammad H. Rezazade Mehrizi, Marleen Huysman, & Frans Feldberg. (2017). “Debating Big Data: A Literature Review on Realizing Value from Big Data.” Journal of Strategic Information Systems 26(3): 191–209.

Blazquez, Desamparados, & Josep Domenech. (2018). “Big Data Sources and Methods for Social and Economic Analyses.” Technological Forecasting and Social Change 130 (March 2017): 99–113. https://doi.org/10.1016/j.techfore.2017.07.027.

Fang, Huang. (2015). “Managing Data Lakes in Big Data Era: What’s a Data Lake and Why Has It Became Popular in Data Management Ecosystem.” 2015 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2015: 820–24.

Khine, Pwint Phyu, & Zhao Shun Wang. (2018). “Data Lake: A New Ideology in Big Data Era.” ITM Web of Conferences 17: 03025.

Miloslavskaya, Natalia, & Alexander Tolstoy. (2016). “Big Data, Fast Data and Data Lake Concepts.” Procedia Computer Science 88: 300–305. http://dx.doi.org/10.1016/j.procs.2016.07.439.

Bell, David, Mark Lycett, Alaa Marshan, & Asmat Monaghan. (2021). “Exploring Future Challenges for Big Data in the Humanitarian Domain.” Journal of Business Research 131(August 2019): 453–68. https://doi.org/10.1016/j.jbusres.2020.09.035.

Kościelniak, H. & Puto, A. (2015). BIG DATA in decision making processes of enterprises. Procedia Computer Science, 65, pp.1052-1058.

Gandomi, Amir, & Murtaza Haider. (2015). “Beyond the Hype: Big Data Concepts, Methods, and Analytics.” International Journal of Information Management 35(2): 137–44. http://dx.doi.org/10.1016/j.ijinfomgt.2014.10.007.

Luckow, Andre et al. (2015). “Automotive Big Data: Applications, Workloads and Infrastructures.” Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015: 1201–10.

Herschel, R. & Miori, V.M., (2017). Ethics & big data. Technology in Society, 49, pp.31-36.

Kim, Y., You, E., Kang, M. & Choi, J., (2012). Does Big Data Matter to Value Creation?: Based on Oracle Solution Case. Journal of Information Technology Services, 11(3), pp.39-48.

Sethi, Pallavi, & Smruti R. Sarangi. (2017). “Internet of Things: Architectures, Protocols, and Applications.” Journal of Electrical and Computer Engineering 2017.

Papazoglou, Michael P., & Amal Elgammal. (2018). “The Manufacturing Blueprint Environment: Bringing Intelligence into Manufacturing.” 2017 International Conference on Engineering, Technology and Innovation: Engineering, Technology and Innovation Management Beyond 2020: New Challenges, New Approaches, ICE/ITMC 2017 - Proceedings 2018-Janua: 750–59.

Sawadogo, Pegdwendé, & Jérôme Darmont.. (2021). “On Data Lake Architectures and Metadata Management.” Journal of Intelligent Information Systems 56(1): 97–120.

Sawadogo, P. N., Scholly, É., Favre, C., Ferey, É., Loudcher, S., & Darmont, J. (2019). Metadata Systems for Data Lakes: Models and Features. Communications in Computer and Information Science, 1064, 440–451. https://doi.org/10.1007/978-3-030-30278-8_43

Beheshti, A., Benatallah, B., Nouri, R. & Tabebordbar, A., (2018). CoreKG: a knowledge lake service. Proceedings of the VLDB Endowment, 11(12), pp.1942-1945.