A Linked Data-based Service for Integrating Heterogeneous Data Sources in Smart Cities

João Gabriel Almeida, Jorge Silva, Thais Batista[a] and Everton Cavalcante[b]

Federal University of Rio Grande do Norte, Natal, Brazil

Keywords: Heterogeneous Data, Data Integration, Linked Data, NGSI-LD, Smart Cities.

Abstract: The evolution and development of new technological solutions for smart cities has significantly grown in recent years. The smart city scenario encompasses larges amount of data from several devices and applications. This raises challenges related to data interoperability, including information sharing, receiving data from multiple sources (Web services, files, systems, etc.), and making them available at underlying smart city application development platforms. This paper presents Aquedücte, a service that converts data from external sources and files to the NGSI-LD protocol, enabling their use by applications relying on an NGSI-LD-based middleware. This paper describes the Aquedücte methodology used to: (i) extract data from heterogeneous data sources, (ii) enrich them according to the NGSI-LD data format using Linked Data along with ontologies, and (iii) publish them into an NSGI-LD-based middleware. The use of Aquedücte is also described in a real-world smart city scenario.

1 INTRODUCTION

In the context of smart cities, a large amount of heterogeneous data is generated, stored (in various formats), and exchanged through different communication protocols. These data sometimes are not readily usable due to the heterogeneity of data types and formats from vertical silos produced by smart city systems (d'Aquin et al., 2015). Making smart cities a reality involves addressing interoperability at data level towards reusing and sharing data. Data interoperability can be achieved through the adoption of a standardized semantic-based data model that unifies the format of data, provides a common meaning to them, and allows for complex reasoning.

Recently, the combination of the Resource Description Framework (RDF)[1], the W3C Web Ontology Language (OWL)[2], and Linked data has been considered the reference practice for sharing and publishing structured data on the Web (Bizer et al., 2009; Consoli et al., 2017). Linked data allows integrating data into a common, browsable, accessible graph, thereby allowing for the use of data across different domains. It has been effective in many cases when information from distinct sources must be put together in a generic way and made available for a variety of applications. By linking data, correlations can be quickly understood. In the smart city context, an exchange protocol named *Next Generation Service Interfaces - Linked Data* (NGSI-LD)[3] has been recently proposed to comply with the Linked Data concept. NGSI-LD defines an information model based on entities, relationships, and properties upon RDF, all of them being semantically represented as concepts of ontologies to ensure homogeneity among data from different sources and contexts.

Even though a standardized data model eases data integration from multiple heterogeneous sources, it is necessary to define an approach that receives such data and converts them to a target semantic-based standardized data model. Aiming at tackling such a challenge, this paper presents Aquedücte, a service that imports data from different sources (third-party Web services or files), converts them to an NGSI-LD-based model, and makes them available to an underlying NGSI-LD-based middleware. Therefore, smart city applications can be built upon that middleware and hence exploit the available shared data.

This paper is structured as follows. Section 2 presents a background about Linked Data and the

[a] https://orcid.org/0000-0003-3558-1450

[b] https://orcid.org/0000-0002-2475-5075

[1] https://www.w3.org/TR/rdf-schema

[2] https://www.w3.org/OWL/

[3] https://bit.ly/2ORxbEV

Almeida, J., Silva, J., Batista, T. and Cavalcante, E.
A Linked Data-based Service for Integrating Heterogeneous Data Sources in Smart Cities.
DOI: 10.5220/0009422802050212
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1*, pages 205-212
ISBN: 978-989-758-423-7

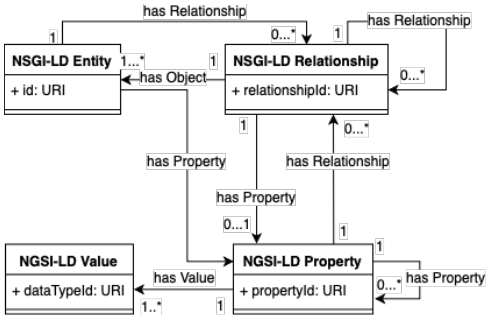


Figure 1: The NGSI-LD Information Model.

NGSI-LD data protocol. Section 3 describes the Aquedücte architecture and its components. Section 4 describes a proof of concept regarding a real-world smart city scenario. Section 5 briefly discusses about related work. Section 6 contains final remarks.

2 BACKGROUND

Linked Data (Heath and Bizer, 2011) is a method to publish interlinked data using standardized Web technologies such as HTTP, RDF, and URIs aiming at building complex information by aggregating simpler information units. Linked Data allows for large-scale integration of data in the Web and reasoning about them. Using RDF, information is represented as a triple <*subject*, *predicate*, *object*> in which an *object* can be another subject, thus allowing interlinking with each other. Such a linking structure forms a directed labeled graph in which edges represent named links between two resources, which are represented by graph nodes.

The ETSI Industry Specification Group for Cross-Cutting Context Information Management (ISG CIM) has recently proposed the NGSI-LD specification, which encompasses an information model with semantic aspects related to Linked Data and ontologies. The main constructs of the NGSI-LD information model are *entities*, *relationships*, and *properties*. An entity represents a real-world object such as a building or a person. A relationship associates entities, e.g., a person working in a building. A property associates values with elements, such as identifying the person name as Alice. Figure 1 depicts the main elements of the NGSI-LD information model.

NGSI-LD entities are represented using JSON-LD[4], an extension of the JSON format now tailored for Linked Data. The JSON-LD aims to serialize entity data in a simple, effective way instead of using RDF triples as commonly adopted in Linked Data. Such an approach is interesting since it takes advantage of the well-known JSON format, thus minimizing possible compatibility issues (Lanthaler, 2013).

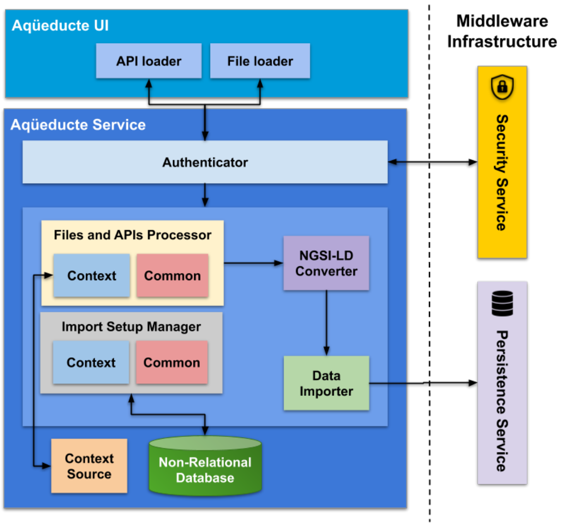


Figure 2: Aquedücte Architecture.

3 THE AQUEDüCTE SERVICE

Aquedücte is a service aimed at standardizing data from external Web sources and files while taking advantage of the facilities and resources provided by the NGSI-LD protocol. The main concern is fostering integration among heterogeneous data by enabling them to be imported to any middleware infrastructure that works with NGSI-LD. Section 3.1 describes the Aquedücte architecture. Section 3.2 presents some implementation details and used technologies.

3.1 Architecture

The Aquedücte architecture is composed of two main elements, namely (i) the *Aquedücte UI* for interacting with users and (ii) the *Aquedücte Service* that implements its main functionalities. Figure 2 provides an overview of the Aquedücte architecture.

The *Aquedücte UI* consists of a user-friendly interface for loading and extracting data from RESTful Web services or files. It consists of two components, *API Loader* and *File Loader*: the former loads data from Web services in the JSON format, whereas the latter handles data in different file formats. This high-level interface allows extracting, filtering, and converting data to the NGSI-LD protocol.

The *Aquedücte Service* provides a RESTful interface and services for NGSI-LD data extraction and conversion operations, besides a non-relational

[4] https://json-ld.org/

database to store import setups created by users. That import setups consist of the parameters that were used to perform data extraction, filtering and importing.

The use of the *Aquedücte Service* requires client authentication provided by the *Authenticator* component, which can use a third-party security service. The main functionalities of the *Aquedücte Service* are realized by four main components, namely (i) *Files and APIs Processor*, (ii) *Import Setup Manager*, (iii) *NGSI-LD Converter*, and (iv) *Data Importer*.

The *Files and APIs Processor* component is responsible for filtering and extracting data through two different approaches, *Context* and *Common*. As means of standardizing extracted data and easing data analysis and queries with NGSI-LD, the *Context* approach takes advantage of LGeoSIM, a semantic information model for smart cities (Rocha et al., 2019). LGeoSIM was chosen as information model as it is able to address data heterogeneity while considering georeferenced information, which can be quite useful for smart city applications. LGeoSIM relies on the ability of defining different layers, each one representing a set of related elements with georeferenced information. Furthermore, LGeoSIM is grounded on the NGSI-LD specification to allow for Linked Data along with ontologies.

The *Common* approach concerns extracting, filtering, and importing data without providing standardization through a specific context source. This approach is valid for cases in which a given data set to be manipulated does not require standardization via a context source from the end-user's point view. Therefore, only using the NGSI-LD protocol would be sufficient.

The *Import Setup Manager* component aims to manage user settings to easing data import. Therefore, users can select one of the registered setups to import data, thus avoiding setting again previously defined parameters.

The *NGSI-LD Converter* component is responsible for converting data either from external Web services or files with different structures. This component handles both common properties (expressed in numeric, text, and collection types) and geolocation properties (expressed in the GeoJSON format[5]) and converts data sets structured as key-value pairs to the NGSI-LD protocol.

Listing 1 shows an example of entity data complying with the NGSI-LD protocol. *id* and *type* are mandatory properties: the former is a unique identifier for the entity and the latter indicates the type represented by the entity. Other properties follow the NGSI-LD standard. In this example, the *location* property is of the *GeoProperty* type and has a GeoJSON value representing geographic information about the entity.

```
{
   "id": "urn:ngsi-ld:Parking:Downtown1",
   "type": "Parking",
   "name": {
      "type": "Property",
      "value": "Downtown One"
   },
   "totalSpotNumber": {
      "type": "Property",
      "value": 200
   },
   "location": {
      "type": "GeoProperty",
      "value": {
         "type": "Point",
         "coordinates": [-8.5, 41.2]
      }
   },
   "@context": [
      "http://uri.etsi.org/ngsi-ld/v1/
       ngsi-ld-core-context.jsonld",
      "http://example.org/ngsi-ld/
       parking.jsonld"
   ]
}
```

Listing 1: Example of NGSI-LD compliant data.

The `@context` field contains links to ontology files that define a data vocabulary for an entity as means of ensuring semantic consistency. For instance, the `@context` field could contain a link pointing to a data vocabulary about vehicles with fields such as *name* and *engine*. Therefore, any entity using this vocabulary must contain at least one of these fields. It is worth highlighting that validating syntactic and semantic information about the entity against the data vocabulary is out of the scope of Aquedücte.

Data Importer is a core component in the Aquedücte architecture. This component is responsible for importing NGSI-LD data sets to an application or middleware able to handle such a protocol. Data import takes place through a RESTful communication between the Aquedücte's *Data Importer* component and an external persistence service.

3.2 Implementation

Aquedücte was developed by following a RESTful service-oriented architecture. The *Aquedücte UI* is a front-end developed with Vuetify[6], a Javascript-based library that provides developers with ready-to-use UI components. This library also comes with two-way

[5] https://geojson.org/

[6] https://vuetifyjs.com/

data binding or reactive approach, which enables declared JavaScript variables to be synchronized with any changes that may occur in the Document Object Model (DOM) or UI.

The *Aquedücte Service* is a back-end developed atop Spring Framework[7], one of the most widely used Java Web frameworks. The Spring Boot module[8] was used to implement the *Aquedücte Service* to ease initial project configuration through the use of dependencies. The Spring Data module[9] was used to persist data at MongoDB, a non-relational database chosen to the *Aquedücte Service* implementation. MongoDB is used by the *Import Setup Manager* component as means of storing import setups, which can be further created and queried by users.

The *NGSI-LD Converter* component consists of RESTful Web services for each approach for converting data to the NGSI-LD protocol, namely *Common* and *Context* (see Section 3.1). In the *Common* approach, the respective service receives an object complying with NGSI-LD as payload, as shown in Listing 2. Such a payload is composed of two fields, *geoLocationConfig* and *dataContentForNGSILDConversion*. The former field constitutes the configurations defined by the user to convert non-GeoJSON geolocation data to the *GeoProperty* type, which has a GeoJSON value. The latter field contains the data to be converted to the NGSI-LD protocol.

```
{
  "geoLocationConfig": [
  {
     "key": "location",
     "typeOfSelection": "string",
     "invertCoords": true,
     "delimiter": ",",
     "typeGeolocation": "Point"
  }],
  "dataContentForNGSILDConversion": [
  {
     "location": "-5.15942190,
        -37.36057650",
     "creDate": "1996-11-12",
     "city": "Natal",
     "schoolName": "Atheneu",
     "size": "Big"
  }, ...
  ]
}
```

Listing 2: Example of payload to convert data to NGSI-LD through the *Common* approach.

The *geoLocationConfig* field has some attributes to help converting geographic data:

- *key* stores the name of the field to be converted to the GeoJSON format;
- *typeOfSelection* stores the type of field (string, collection or GeoJSON) to be converted to GeoJSON;
- *invertCoords* is a Boolean attribute that indicates if the order of latitude and longitude values need to be inverted when the *typeOfSelection* attribute is different from GeoJSON;
- *delimiter* consists of an attribute that stores a delimiter (pipe, comma or any other) when the *typeOfSelection* attribute is of the string type;
- *typeGeolocation* stores a GeoJSON data type (point, polygon), being relevant only when the *typeOfSelection* attribute is different from GeoJSON as data formatted in GeoJSON already comes with this attribute type.

The *Context* approach differs from the *Common* approach since the payload received by the service contains two additional properties as means of maintaining semantic consistency. Listing 3 shows how such a payload is structured.

The *contextLink* property represents a JSON-LD context file selected by the user with a data vocabulary, which is used to standardize data from different sources. This standardization is done through a matching process in which the properties available at the context file loaded by the user will be matched with the ones derived from data extracted available at external sources. This mapping generates the *matchingConfigContent* property with the following fields:

- *contextName* receives the property/attribute of the loaded context file;
- *foreignProperty* receives the property/attribute from data extracted from external sources;
- *isLocation* is a Boolean field that indicates if a certain property derived from the extracted data should be converted to GeoJSON, following the settings defined by the *geoLocationConfig* field;
- *geoLocationConfig* has attributes to help converting geographic data.

4 PROOF OF CONCEPT

The Aquedücte service is currently integrated to *Smart Geo Layers* (Souza et al., 2018), a smart city middleware platform aimed to (i) integrate data provided by heterogeneous sources, (ii) support data correlation with geographic information, and (iii) provide functionalities such as data aggregation, visualization, querying, and analysis. In terms of security, Aquedücte uses the authentication and persis-

[7] https://spring.io

[8] https://spring.io/projects/spring-boot

[9] https://spring.io/projects/spring-data

```
{
   "contextLink":
      "https://url.com/ngsi-ld/education/
      school/School_Context.jsonld",
   "matchingConfigContent": [
   {
      "contextName": "description",
      "foreignProperty": "schoolName",
      "isLocation": false,
      "geoLocationConfig": {}
   },
   {
      "contextName": "city",
      "foreignProperty": "schoolCity",
      "isLocation": false,
      "geoLocationConfig": {}
   },
   {
      "contextName": "location",
      "foreignProperty": null,
      "isLocation": true,
      "geoLocationConfig": [
      {
         "key": "geoLocation",
         "typeOfSelection": "string",
         "invertCoords": true,
         "singleFieldLocation": "",
         "delimiter": ",",
         "typeGeolocation": "Point"
      }]
   }],
   "dataContentForNGSILDConversion": [
   {
      "geoLocation": "-5.15942190,
         -37.36057650",
      "creDate": "1996-11-12",
      "schoolCity": "Natal",
      "schoolName": "Atheneu",
      "schoolSize": "Big"
   },
   ...
   ]
}
```

Listing 3: Example of payload to convert data to NGSI-LD through the *Context* approach.

tence services provided by *Smart Geo Layers* as middleware infrastructure.

The main validation use case was carried out at the Public Prosecution Service of Rio Grande do Norte (MPRN), Brazil. Aquedücte was used in the process of extracting, filtering, and importing data using a Web service provided by MPRN. Data consist of geographical coordinates in GeoJSON format and the Basic Education Development Index (IDEB) of each municipality of the state of Rio Grande do Norte. Once already imported into *Smart Geo Layers*, these data can be queried through a Web application built atop the middleware to display them at a map with each municipality and its corresponding IDEB.

The steps for importing data into *Smart Geo Layers* (see Figures 3 and 4) are:

1. Select the import setup type: *Context* or *Common* (default) approach
2. Select the type of data extraction of import setup: from a file or from external API/Web service
3. Select the domain layer which each imported entity will belong to
4. Setup for request to an external Web service and data set loading
5. Select data from source to be handled
6. Filter fields for importing
7. Select fields for GeoJSON conversion (optional);
8. Convert to NGSI-LD protocol and visualize the converted data
9. Import data into middleware by using the persistence service

Figure 5 presents a Web application with the plot of each city in the state of Rio Grande do Norte with its respective educational indices as the final result of the importation process using Aquedücte. The use of Aquedücte was essential for using data/information from external sources by *Smart Geo Layers*. As it converted the imported data to the NGSI-LD protocol, they are available to applications that use any NGSI-LD-based middleware.

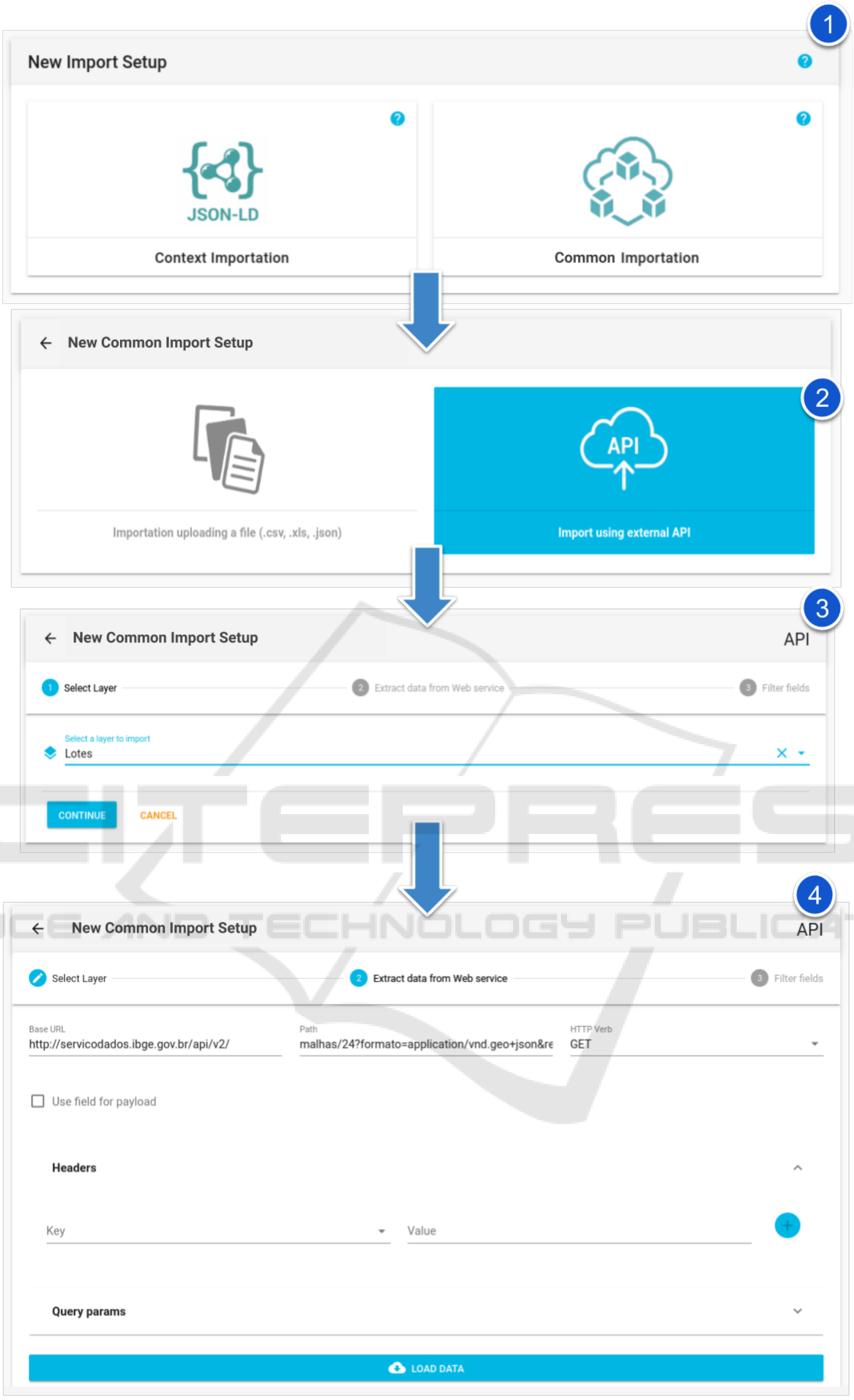
5 RELATED WORK

We conducted a systematic literature review on the extraction, integration, and data import within the scope of smart cities. Major publication electronic databases such as IEEEXplore, ACM Digital Library, Scopus, ScienceDirect.com, and Web of Knowledge were used to automatically retrieve studies. The following search string was used:

```
("data integration" OR "data extraction"
OR "data importation")
AND ("smart city" OR "smart cities")
```

The search process returned 47 studies. Most of the selected works address data integration, extraction or import aiming to meet a specific domain in smart cities, such as transport, environment, and safety. Three of them have drawn attention due to some similarities with the proposal of Aquedücte.

(Fortini and Davis, 2018) address the context of urban transportation. The extraction/collection of urban data provided by heterogeneous sources allows the visualization of indicators and geometry of a given city. For example, it allows comparing and evaluating both public and private transport efficiency indicators.



Select a layer to import: Lotes

Base URL: http://servicodados.ibge.gov.br/api/v2/

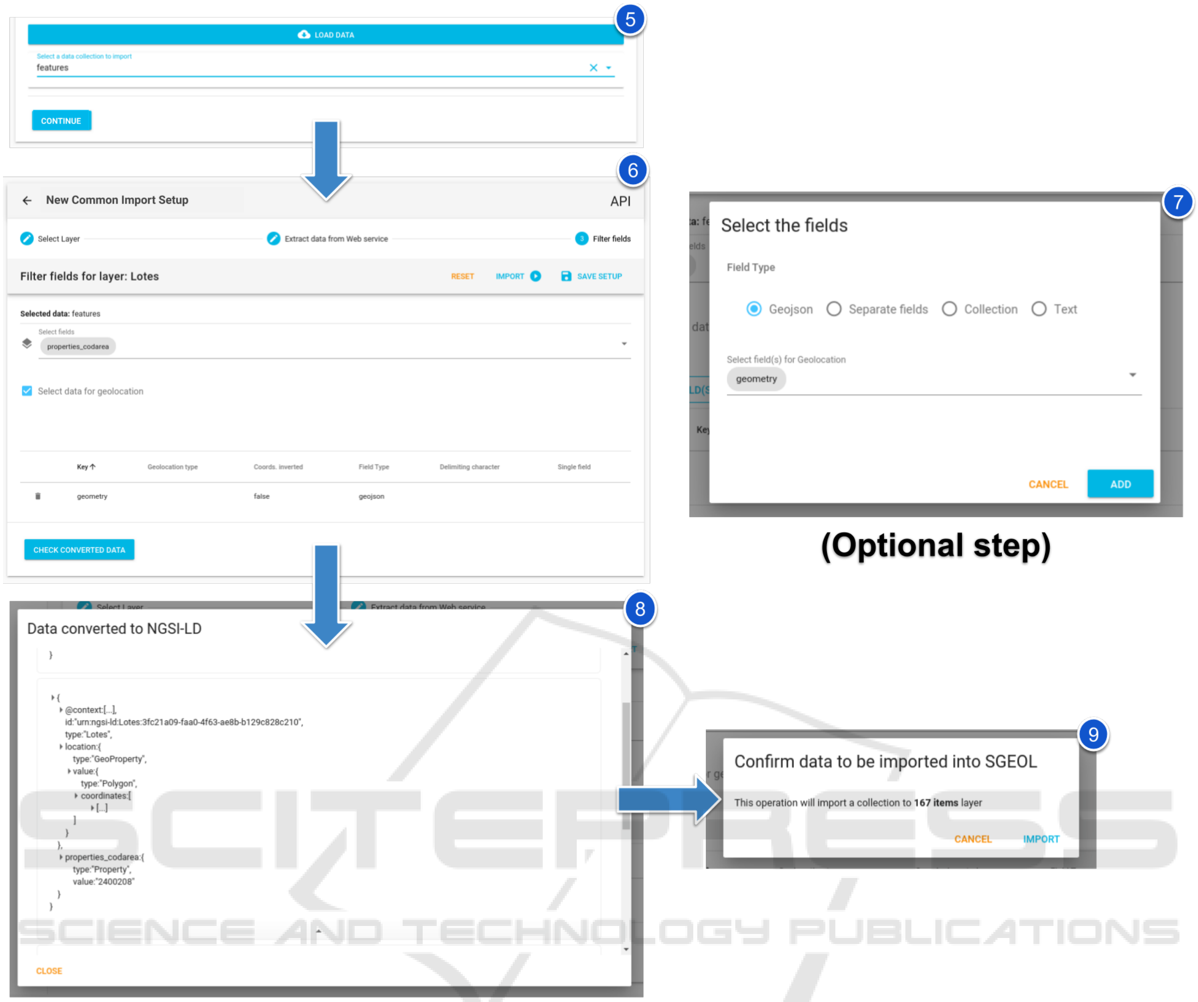
Path: malhas/24?formato=application/vnd.geo+json&re

HTTP Verb: GET

Figure 3: Aquedücte UI Workflow (Part 1/2).

(Mehmood et al., 2019) propose a data lake approach, which would be supplied by diverse sets of data from four pilot cities based on Big Data technologies, e.g., Hadoop File System, Spark, and Apache Flume. From this data lake, it would be possible to analyze and visualize its data, thus providing more efficient decision making in smart cities.

SmartLand-LD (Piedra and Suárez, 2018) is a Linked Data-based framework that provides a flexible distributed ecosystem for data collection, extraction, and publication. It proposes an infrastructure to achieve interoperability and data integration from di-



5

LOAD DATA

Select a data collection to import: features

CONTINUE

6

New Common Import Setup — API

Select Layer — Extract data from Web service — 3 Filter fields

Filter fields for layer: Lotes — RESET — IMPORT — SAVE SETUP

Selected data: features

Select fields: properties_codarea

☑ Select data for geolocation

	Key ↑	Geolocation type	Coords. inverted	Field Type	Delimiting character	Single field
	geometry		false	geojson		

CHECK CONVERTED DATA

7

Select the fields

Field Type

● Geojson ○ Separate fields ○ Collection ○ Text

Select field(s) for Geolocation: geometry

CANCEL — ADD

(Optional step)

8

Data converted to NGSI-LD

```
}

{
  @context:[...],
  id:"urn:ngsi-ld:Lotes:3fc21a09-faa0-4f63-ae8b-b129c828c210",
  type:"Lotes",
  location:{
    type:"GeoProperty",
    value:{
      type:"Polygon",
      coordinates:[
        [...]
      ]
    }
  },
  properties_codarea:{
    type:"Property",
    value:"2400208"
  }
}
```

CLOSE

9

Confirm data to be imported into SGEOL

This operation will import a collection to **167 items** layer

CANCEL — IMPORT

Figure 4: Aquedücte UI Workflow (Part 2/2).

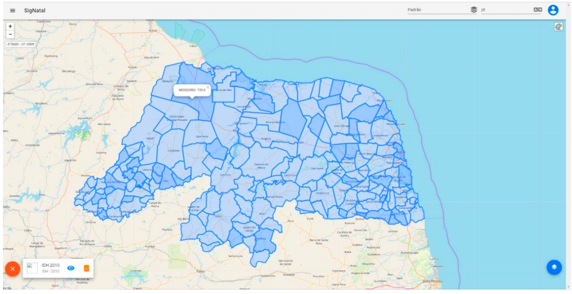


Figure 5: A Web Application That Uses NGSI Converted Data from *smart Geo Layers*.

verse sources that may exist in a smart city through the use of ontologies. This concept is applied through the conversion of the extracted data into the RDF format. The framework also provides data/resources availability through a well defined API to end-users and several other applications.

These works have some points in common with Aqueducte in terms of addressing data integration of different formats and sources. However, the main differences are (i) the use of the NGSI-LD protocol to ease the integration of heterogeneous data and (ii) the provision of a friendly UI that makes data extraction, filtering, and import process easier for any middleware or platform that works with the NGSI-LD protocol, as shown in the use case of MPRN (see Section 4).

6 FINAL REMARKS

This paper presented Aquedücte, a service for handling heterogeneous data integration in smart city applications. Aquedücte supports data extraction, filtering, and import for any middleware that works with the NGSI-LD protocol, to which data from different formats and sources can be converted. Aquedücte was initially validated in a real-world scenario in conjunction with an NGSI-LD-compliant middleware platform that allows integrating data provided by heterogeneous sources, correlating them to geographic information, and aggregating visualizing, querying, analyzing these data.

Ongoing work is currently focused on improvements on Aquedücte to support loading of large files to have data extracted, filtered, and converted to the NSGI-LD format. In addition, Aquedücte will operate together with a microservice to perform user-defined relationships of data to be imported, following the NGSI-LD specification. Such features will be possible due to the adoption of a distributed file system, which will allow managing such a large volume of data in a more effective way.

REFERENCES

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked Data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22.

Consoli, S., Presutti, V., Recupero, D. R., Nuzzolese, A. G., Peroni, S., Mongiovi', M., and Gangemi, A. (2017). Producing Linked Data for smart cities: The case of Catania. *Big Data Research*, 7:1–15.

d'Aquin, M., Davies, J., and Motta, E. (2015). Smart cities' data: Challenges and opportunities for semantic technologies. *IEEE Internet Computing*, 19:66–70.

Fortini, P. M. and Davis, C. A. (2018). Analysis, integration and visualization of urban data from multiple heterogeneous sources. In *Proceedings of the 1st ACM SIGSPATIAL Workshop on Advances on Resilient and Intelligent Cities*, page 17–26, New York, NY, USA. ACM.

Heath, T. and Bizer, C. (2011). *Linked Data: Evolving the Web into a global data space*. Morgan & Claypool Publishers.

Lanthaler, M. (2013). Creating 3rd Generation Web APIs with Hydra. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 35–38, New York, NY, USA. ACM.

Mehmood, H., Gilman, E., Cortes, M., Kostakos, P., Byrne, A., Valta, K., Tekes, S., and Riekki, J. (2019). Implementing Big Data lake for heterogeneous data sources. pages 37–44, USA. IEEE.

Piedra, N. and Suárez, J. P. (2018). SmartLand-LD: A Linked Data approach for integration of heterogeneous datasets to intelligent management of high biodiversity territories. In Mejia, J., Muñoz, M., Rocha, Á., Quiñonez, Y., and Calvo-Manzano, J., editors, *Trends and Applications in Software Engineering*, volume 688 of *Advances in Intelligent Systems and Computing*, pages 207–218. Springer International Publishing AG, Switzerland.

Rocha, B., Cavalcante, E., Batista, T., and Silva, J. (2019). A Linked Data-based semantic information model for smart cities. In *Proceedings of the IX Brazilian Symposium on Computing Systems Engineering*, USA. IEEE.

Souza, A., Pereira, J., Batista, T., Cavalcante, E., Cacho, N., Lopes, F., and Almeida, A. (2018). A geographic-layered data middleware for smart cities. In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, pages 411–414, New York, NY, USA. ACM.