

Wat€rnomics

D3.1.1 Linked Water Dataspace

Project Acronym: **Waternomics**
Project Title: **ICT for Water Resource Management**
Project Number: **619660**
Instrument: **Collaborative project**
Thematic Priority: **FP7-ICT-2013.11**

D3.1.1

Work Package:	WP3	
Due Date:	31/12/2014	
Submission Date:		
Start Date of Project:	01/02/2014	
Duration of Project:	36 Months	
Organisation Responsible of Deliverable:	NUIG	
Version:	1.3	
Status:	Draft	
Author name(s):	Daniele Bortoluzzi Edward Curry Wassim Derguech Souleiman Hasan Fadi Maali Diego Reforgiato Arkadiusz Stasiewicz Umair Ul Hassan	R2M NUIG NUIG NUIG NUIG R2M NUIG NUIG
Reviewer(s):	Christos Kouroupetroglou Peter O'Donovan Schalk-Jan van Andel Nick van de Giesen	Ultra4 NUIG UNESCO-IHE TU Delft
Nature:	<input checked="" type="checkbox"/> R – Report <input type="checkbox"/> P – Prototype <input type="checkbox"/> D – Demonstrator <input type="checkbox"/> O – Other	
Dissemination level:	<input checked="" type="checkbox"/> PU – Public <input type="checkbox"/> CO – Confidential, only for members of the consortium (including the Commission) <input type="checkbox"/> RE – Restricted to a group specified by the consortium (including the Commission Services)	
Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		

Revision history

Version	Date	Modified by	Comments
0.1	22/07/14	Edward Curry	Initial Version of Document
0.2	18/12/2014	Wassim Derguech	Add/Edit section minimal vocabulary
0.2	Dec 2014	Wassim Derguech	Dataspace functional requirements
0.3	Dec 2014	Souleiman Hassan	Linked Data
0.3	Dec 2014	Umair Ul Hasan	Entity Management
0.3	Jan 2015	Souleiman Hasan	Real-time data
0.3	Jan 2015	Arkadiusz Stasiewicz	OpenCube
0.4	Jan 2015	Christos Kouroupetroglou	Data Inputs from WP4
0.4	Jan 2015	Wassim Derguech	Review + executive summary
0.5	Jan 2015	DomenicoPerfido	Appendix C – Survey of Open Data for Water management
0.6	Jan 2015	Peter O'Donovan	Review comments
0.6	Jan 2015	Schalk Jan van Anandel	Review comments
0.7	Jan 2015	Wassim Derguech	Updated Deliverable based on Reviews
0.8	Jan 2015	Wassim Derguech	change conclusion of section 5 and remove appendix B
1.0	Jan 2015	Wassim Derguech	Final version 1.0
1.1	Feb 2015	Wassim Derguech	Update executive summary and introduction
1.2	Feb 2015	Wassim Derguech	Add section “About Waternomics”
1.3	Feb 2015	Souleiman Hasan	Update Section 5 and 6 to match latest findings after the February Hackathon
1.4	Mar 2015	Wassim Derguech	Update based on reviews

Copyright © 2015, Waternomics Consortium

The Waternomics Consortium (<http://www.waternomics.eu/>) grants third parties the right to use and distribute all or parts of this document, provided that the Waternomics project and the document are properly referenced.

THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Executive Summary

This deliverable reports on the current efforts towards designing a Linked Water Dataspace. A dataspace is an emerging information management approach used to tackle heterogeneous data sources that supports requirements, such as standardization, enrichment, and linking of data in an incremental manner.

Designing and implementing a dataspace is the core of the Waternomics information platform and is one of the major objectives for the Waternomics project. In the Waternomics vision collecting, standardizing, enriching and linking water usage data coming from sensors together with other relevant contextual data sources is a key step needed for effective analytics to drive decision making: e.g., planning, adjustments and predictions and to raise user awareness of water consumption.

In this report, we detail the current design and implementation of the dataspace using Linked Data as a recommended W3C best practice for exposing, sharing and connecting pieces of data, information and knowledge. The deliverable introduces the concept of Linked Data, Semantic Web and Web of Data and how these are applied in the context of the Waternomics dataspace.

The primary contribution of this report is a customized architecture for implementing a Linked Water Dataspace. The deliverable exposes a first version of a water data ontology used for semantically describing sensors and their data. The design of the ontology is informed of by the requirements of the pilots for importing or exporting data from/to the dataspace.

Table of Contents

Executive Summary	5
Table of Contents	6
List of Figures	8
List of Tables	10
List of Listings	11
1 Introduction	12
1.1 Work Package 3 Objectives	12
1.2 Purpose and Target Group of the Deliverable	12
1.3 Relations to other Activities in the Project	13
1.4 Document Outline.....	14
1.5 About Waternomics	14
2 Functional Dataspace Requirements	16
3 Linked Data	18
3.1 Overview	18
3.2 The Semantic Web.....	19
3.3 Web of Data	21
3.4 Linked Water Data.....	23
3.5 Conclusion	24
4 State of the Art Analysis	25
4.1 Data Adapters	25
4.2 Linked Data Platforms	28
4.2.1 Virtuoso.....	28
4.2.2 LinDA	29
4.2.3 Pubby.....	29
4.2.4 Apache Marmotta.....	30
4.2.5 LDP4j.....	30
4.2.6 OpenCube Toolkit	31
4.2.7 LOD2 Linked Data Stack	32
4.2.8 CKAN.....	33
4.3 Entity Management	33
4.3.1 Master Data Management	34
4.3.2 Crowdsourced Entity Management.....	35
4.3.3 Collaborative Entity Management	36
4.4 Lambda Architecture	37
4.4.1 DRUID Platform	37
4.4.2 Apache Spark	38
4.5 Conclusion	39
5 A Realtime Linked Water Dataspace	41
5.1 Dataspace Overview	43
5.2 The Dataspace CATALOG Service	45
5.2.1 Managed Entities	47

5.2.2	Information Sources.....	48
5.3	The Dataspace QUERY Service and the Realtime Aspect	49
5.3.1	Data Cube Vocabulary.....	52
5.3.2	Data quality and Data Linking.....	52
5.3.3	Data exploration.....	53
5.3.4	OpenCube for Water Data	53
5.3.5	Data Publishing.....	53
5.3.6	Data Reuse.....	54
5.4	Joining the Dataspace and Adapters.....	55
5.5	Conclusion and Plan	56
6	Concrete Cases	59
6.1	The Building Management System Joins the Dataspace	59
6.2	A Water Sensor Joins the Dataspace.....	60
7	Summary.....	63
	Appendix A: Minimal Vocabulary	64
	Appendix B: Survey of Open Data for Water Management	76
8	References	82

List of Figures

Figure 1: Relationships between D3.1.1 and other activities in Waternomics	14
Figure 2: Relationships between identifier, resource, and representation [7]	18
Figure 3: Basic HTTP architecture	19
Figure 4: An example HTML Web page	19
Figure 5: The Semantic Web layer cake diagram [9]	20
Figure 6: An example RDF graph [11]	20
Figure 7: An example RDF N3 serialization [11]	20
Figure 8: An example SPARQL query	21
Figure 9: Content negotiation in Linked Data [11]	22
Figure 10: 5 stars scheme to publish Linked Open Data on the Web [3]	22
Figure 11: Dereferencing http://dbpedia.org/resource/Water [15]	22
Figure 12: Linked Open Data cloud [16]	23
Figure 13: Ultrawrap architecture	25
Figure 14: Google Refine RDF extension user interface1	26
Figure 15: mapping tabular data to RDF using Grafter DSL2	27
Figure 16: Sample SPARQL query in TARQL	27
Figure 17: Equivalent SPARQL query	27
Figure 18: Linked Data Platform types	28
Figure 19: Virtuoso Architecture Diagram2	29
Figure 20: Pubby Architecture Diagram1	30
Figure 21: OpenCube Toolkit sample page	31
Figure 22: Statistical Workbench dashboard	32
Figure 23: CKAN dashboard	33
Figure 24: Illustration of master data management based solution for entity data management	34
Figure 25: Relative volumes of data relevant to an enterprise, where MDM manages small part of data available in internal and external sources	35
Figure 26: In continuum of entity management approaches the number of contributors increase from top-down to bottom-up. A hybrid approach would help expand boundaries of enterprise entity management	36
Figure 27: Community collaboration based approach for management of entity data	36
Figure 28: An architectural overview of the DRUID [56]	37
Figure 29: The Apache Spark family [44]	38
Figure 30: Querying different data sources with Spark SQL [44]	38
Figure 31: Integrating Spark SQL with Hive [44]	38
Figure 32: Standard data connectivity [44]	39
Figure 33: Standard data connectivity [44]	39
Figure 34: Spark cluster deployment [44]	39
Figure 35: System Architecture	41
Figure 36: Linked Water Dataspace Overview- Components View	43
Figure 37: Linked Water Dataspace- The CATALOG Service View	46
Figure 38: Waternomics Dataspace – Data sets related to Galway	46
Figure 39: NEB Historical Logs in Waternomics dataspace	47
Figure 40: Linked Water Dataspace Lambda Serving Layer-- The QUERY Service View	50
Figure 41: Outline of the Data Cube vocabulary	52
Figure 42: TARQL Data Provider configuration screen	54
Figure 43: OpenCube Chart Visualization widget	54
Figure 44: RETLEMM process for exposing entities as Linked data	55
Figure 45: Linked Water Dataspace Lambda Architecture Realization	59
Figure 46: An example realization of Lambda Batch Layer in Linked Water Dataspace	60
Figure 47: Linked Water Dataspace Lambda Speed Layer	61
Figure 48 – Phase 1 and 2 of the design of the RDF minimal vocabulary for Waternomics	64
Figure 49 – Phase 3 of the design of the RDF minimal vocabulary for Waternomics	65
Figure 50 – Entity Relationship Model: Sensor	66
Figure 51 – RDF graph example for Sensor	67
Figure 52 – Entity Relationship Model: Observation_Type	67
Figure 53 – RDF graph example for Observation_Type	69
Figure 54 – Entity Relationship Model: Sensor_Observation_Type captured as Observes relation	69
Figure 55 – RDF graph example for Sensor_Observation_Type	71
Figure 56 – Entity Relationship Model: Observation	71

Figure 57 – RDF graph example for Observation	73
Figure 58 –Entity Relationship Model: Aggregated_Observation	73
Figure 59 – RDF graph example forAggregated_Observation	75
Figure 60: GEOSS logo[47]	77
Figure 61: GEOSS Portal [47]	77
Figure 62: RECODE logo	78
Figure 63: Look and feel of Vowl.	78
Figure 64: Screenshot of the WISDOM project website	79
Figure 65: <i>Architecture of the sMAP project</i>	80
Figure 66: <i>Pachube old website</i>	81
Figure 67: <i>Architecture of the SMART Knowledge Base [61]</i>	81

List of Tables

Table 1: Analysis of the State of the Art	40
Table 2: Proposed Approaches for the Dataspace	41
Table 3: Linked Water Dataspace Support Services and Requirements	44
Table 4: Required attributes for the entity Sensor	48
Table 5: Required attributes for the entity Outlet.....	48
Table 6: Required attributes for the entity Site	48
Table 7: Summary of Requirements and Approaches.....	57
Table 8: Mapping from entity relationship item to RDF concepts	65
Table 9: Mapping Sensor Entity Relationship Model to RDF	66
Table 10: Sensor Table Example	66
Table 11: Mapping Observation_Type Entity Relationship Model to RDF	67
Table 12: Observation_Type Table Example	68
Table 13: Mapping Sensor_Observation_Type Entity Relationship Model to RDF.....	69
Table 14: Sensor_Observation_Type Table Example.....	70
Table 15: Mapping Observation Entity Relationship Model to RDF	72
Table 16: Observation Table Example	72
Table 17: Mapping Aggregated_Observation Entity Relationship Model to RDF.....	73
Table 18: Aggregated_Observation Table Example.....	74

List of Listings

Listing 1: Sensor JSON Data	61
Listing 2: Sensor Dimensional JSON Data	62
Listing 3: Sensor Example RDF n3	66
Listing 4: Observation_Type Example RDF n3	68
Listing 5: Sensor_Observation_Type Example RDF n3	70
Listing 6: Observation Example RDF n3	72
Listing 7: Aggregated_Observation Example RDF n3.....	74

1 Introduction

The goal of Waternomics is to explore how ICT can help households, businesses and municipalities with reducing their consumption and losses of water. A key component of the Waternomics information platform aims at collecting water consumption and contextual information from different sources to be used for effective data analytics to drive decision making: e.g., planning, adjustments and predictions and to raise user awareness of water consumption.

A key outcome of the work carried out in Work Package 3 consists of designing a Linked Water Dataspace. A dataspace is an emerging information management approach used to tackle heterogeneous data sources and that supports requirements, such as standardization, enrichment, and linking of data in an incremental manner. The primary contribution of this report is a customized first version of architecture for implementing such a Linked Water Dataspace. This architecture will be further refined depending on the other tasks and work packages requirements. The final version of the Linked Water Dataspace will be reported in D3.1.2.

1.1 Work Package 3 Objectives

The objective of Work Package 3 (WP3) is to develop the project software and user environment that will deliver water information services to various targeted stakeholders. It allows linking sensors, data management systems and water meters. More specifically, the objectives of WP3 are as follows:

- To develop a linked dataspace able to capture and store data from various sources
- To develop a set of services based on the web as a platform allowing applications to interact and use the data stored in the linked dataspace
- To provide a set of applications consisting of various customisable and personalised components
- To deploy and customize the developed applications in appropriate sites for validation and evaluation

WP3 will receive as input the high-level system architecture, usage and exploitation cases, and KPIs for reporting from WP1. WP3 will provide as output the water information services platform for the pilots and future exploitation as a primary project outcome/result. In particular, WP3 will develop and provide user friendly content, applications, and platforms that enable all stakeholders (utilities, commercial users and domestic users) to take decisions that result in reduced water consumption and losses, and increase overall awareness of drinking water supply issues.

1.2 Purpose and Target Group of the Deliverable

The objective of this deliverable is to report on the progress of Task 3.1 (Data sources adapters customization / development) and 3.2 (Linked water data infrastructure design). Both tasks contribute to the management of a dataspace containing water data that is later used by support services for dedicated applications. We call this dataspace the **Linked Water Dataspace**. This data space will be further refined with respect to the project evolution and a final version will be available in D3.1.2.

A **Linked Water Dataspace** is a central technological concept of the Waternomics information platform. Its role consists of providing an interoperability space with common access mechanisms to the data. Important components of the dataspace are the data sources adapters that take as input sensors and other sources of data and provide a linked data-cloud rich with knowledge and semantics about the water consumption. These adapters require a predefined semantic model for describing both sensor data and meta-data.

The **semantic model for sensor data** is a formal ontology developed within the Waternomics project. This model has been designed with respect to the data models used by existing management systems.

Ontologies and adapters of the Waternomics project are proposed in order to:

- Create a linked data cloud,
- Facilitate data integration across multiple platforms,
- Facilitate data access and interoperability.

The main target groups for this deliverable are designers of water management systems as well as ontology designers and technicians. The minimal vocabulary proposed in this document is also of interest to domain experts in water management.

1.3 Relations to other Activities in the Project

Figure 1 illustrates the relations of this deliverable to other activities in the Waternomics project. These relations are represented as links numbered from 1 to 4 and are described as follows:

Link 1: The design of the Linked Water Dataspace follows the set of dataspace requirements identified in Section 2.4.2 of the Deliverable D 1.3. These requirements are further refined and explained in Section 2.

Link 2 and 3: This deliverable reports on the design of the Linked Water Dataspace. This feeds into D3.2 (Support Services APIs and Components Libraries) and D3.3 (Waternomics Apps) as both require the technical details of the dataspace for the developments of support services and applications.

Link 4: Pilot planning in WP5 also uses this deliverable as it constitutes a detailed background analysis and technical design work towards the Linked Water Dataspace.

Link 5: A revised version of the Linked Water Dataspace proposed in this deliverable (D3.2.1) will be detailed in D3.1.2.

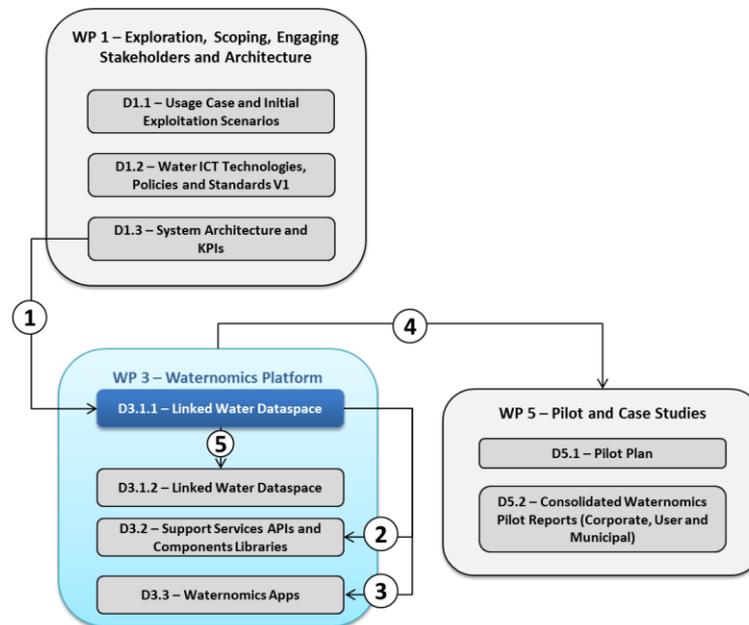


Figure 1: Relationships between D3.1.1 and other activities in Waternomics

1.4 Document Outline

The remainder of this document is organised as follows:

- Section 2: Functional Dataspace Requirements – defines the set of requirements for building the Linked Water Dataspace.
- Section 3: Linked Data – introduces the concepts of semantic web and web of Data and how it can be applied in the context of linked water data.
- Section 4: State of the Art Analysis – discusses relevant contributions that can be used for designing and implementing the Waternomics dataspace.
- Section 5: A Realtime Linked Water Dataspace- defines the proposed architecture for managing water data and exposing it into the Linked Water Dataspace.
- Section 6: Concrete Cases- presents two concrete cases for datasets joining the dataspace, being the NUIG engineering building's BMS, and the VTEC water sensor.
- Section 7: Summary- concludes the deliverable.
- Appendix A: Minimal Vocabulary – maps entity relationship model of D1.3 to ontological concepts as a proposal for a minimal vocabulary for modelling sensors and their observations
- Appendix B: Survey of Open Data for Water Management – investigates relevant contributions that are using open data for water management.

1.5 About Waternomics

Climate change, increased urbanization and increased world population are several of the factors driving global challenges for water management. In fact, the World Economic Forum has cited “The Water Supply Crises” as a major risk to global economic growth and environmental policies in the next 10 years. In parallel, the United Nations has called for intensified international collaboration. To help reduce water shortages, Waternomics will explore the technologies and methodologies needed to successfully reduce water consumption and losses

from households, companies and municipalities. Waternomics is a three year EU-funded project that started in February 2014 that will develop and introduce ICT as an enabling technology to manage water as a resource, increase end-user conservation awareness and affect behavioural changes, and to avoid waste through leak detection. In saving water, energy will also be conserved (treatment and pumping) as will the CO₂ associated with energy production. Unique aspects of WATERNOMICS include personalized feedback about end-user water consumption, the development of a methodology for the design and implementation of systematic and standards-based water resource management systems, new sensor hardware developments to make water metering more economic and easier to install, and the introduction of forecasting and fault detection diagnosis to the analysis of water consumption data.

WATERNOMICS will be demonstrated in three high impact pilots that target three different end users/stakeholders:

- Domestic users in Greece implemented by a water utility
- Corporate operator in Italy provided by a major EU airport
- Public and Mixed-use based demonstration in Ireland

Through these contributions, WATERNOMICS will pioneer a new dialogue between water stakeholders. It will enable the introduction of Demand Response principles and open business models through an innovative human centric approach that uses personalized water data, water availability based pricing, and gamification of water usage statistics. To maximize impact, the project highlights business development, exploitation planning, and outcome oriented dissemination.

2 Functional Dataspace Requirements

Deliverable D1.3- Section 2 identified the functional requirements as well as the set of generic architecture functions in that the Linked Water Dataspace should respect when dealing with input data collected from sensors and other data sources. These requirements also specify how the output data is made available to support services and applications. In this section, we carry out a further analysis of these requirements and propose the following list of functional requirements for the dataspace:

- **Standardisation:** The system will serve as a common dataspace for different stakeholders. Consequently, the data exchanged and published by the system should be standardised. The same data and support services will be available to all applications. With respect to this requirement we propose to use RDF for describing the various entities managed within the system. An analysis of existing RDF data adapters is reported in Section 4.1. Details about the proposed RDF vocabulary and its alignment with existing ontologies are described in details in Appendix A: Minimal Vocabulary.
- **Consuming Open Data:** By definition Open Data “*can be freely used, modified, and shared by anyone for any purpose*” [1]. In our context, the system should be able to make use of relevant open data assets for proper analytics. Possible scenarios for consuming open data include the prediction of water consumption using open weather data. Consuming open data requires a proper selection and evaluation of data source in order to select the most suitable one for proper decision support [2]. An initial survey of Open Data sources for water management are presented in Appendix B: Survey of Open Data for Water Management
- **Publishing Linked Data:** The data produced by adapters or support services should be published in the dataspace with respect to linked data principles¹: available on the web, structured, not using a proprietary format, using Uniform Resource Identifiers (URIs) to denote entities and linked to other data sets [3]. Details about these principles will be discussed in Section 3 and Section 5.3.
- **Data Linking:** When publishing water data to the dataspace, it has to be linked to other data sets. This linking is very useful for ensuring an optimal data management and integration. It helps enhancing their (re)use and discovering new knowledge from water data put into a wider context. It is important to assess and determine what data sets are relevant to be linked with water data. Section 5.3.2 provides further details about this requirements and how it is covered by the dataspace.
- **Real-time data / events:** The system will be handling continuous streams of data coming from multiple sensors and data sources. The system should be able to manage large quantities of data in real-time. Real-time processing of data requires the development of algorithms and tools for parallel processing of simple and complex events (see Section 6.2).
- **Real-time Analytics:** Data analytics needs to be continuously made in order for consumers’ applications to make timely decisions i.e. the analytics should make use of the latest data available. Speed layer from the Lambda architecture is covering this requirement. Details are given in Section 6.2.
- **Data integration:** The system analytics should integrate both real-time and historical data for effective decision support. Therefore, the system should be capable of seamlessly

¹<http://5stardata.info/>

analysing water consumption and provide integrated view of the data being analysed. The use of Lambda Architecture covers this requirement as in essence; it facilitates the integration of real-time (speed layer) and historical data (batch layer). For a discussion of existing implementations of the Lambda architecture, we refer to Section 4.4.

- **Heterogeneity of Sensor Data Events:** The system will be handling data from a wide variety of sensors and consequently a wide variety of data formats. The dataspace needs to be able to handle applications' queries across data formats with respect to their semantic similarity. Additionally, the dataspace should manage data produced by developed services from other work packages such as leakage detection data from WP4.
- **Enrichment of Sensor Data Events with Open Data:** Raw sensor data reports mainly on the observed values of a particular property. This data requires additional contextual information, such as the location of the sensor, in order to deliver accurate decision analytics. The dataspace needs to enrich sensor data with relevant information that are required by support services and applications.

In order to cover these requirements, we propose to use Linked Data as a set of best practices for publishing, sharing and interlinking structured data on the web. In such context, if all the data of our infrastructure is open and linked to other open data sets from the web, it would be easier to create a water information system combining various distributed data repositories. Thus this would enable access and sharing of water data without barriers for building effective services and application.

3 Linked Data

In the following we give an account of the evolution towards Linked Data. That evolution can be traced back to the early days of the Web and then with the realisation that the Web has to be extended with structured data and semantics giving birth to the Semantic Web. Linked Data then appeared within the Semantic Web realm to give more attention to the importance of publishing data on the Web and how the Semantic Web technologies can serve that. Finally, this section discusses how water data can benefit from this technology.

3.1 Overview

The World Wide Web has been conceived and first implemented in CERN by Tim Berners Lee in 1989 [4]. Web builds upon the Internet network and uses a system of hypertext documents interlinked together via hyperlinks. A piece of information in such a system is the Web document which may contain text, images, videos and other multimedia content. Web documents, or pages, are typically hosted by Web servers which can be queried to retrieve the page.

Typically, a web page can be abstracted and looked at as a resource, which may or may not exist in the real-world. For example, Figure 1 illustrates relationships of a set of concepts that form basic concepts in the Web architecture. Resource in Figure 1 is a weather report, which has an identifier represented by the Web standard identification system, namely the Unified Resource Identifier (URI) [5] which is “http://weather.example.com/oaxaca” in this case. The resource is still not an information resource but rather just a resource. Nonetheless, it can be captured in information systems, e.g. the Web architecture, using an information resource such as a textual document. The Web uses the HyperText Markup Language (HTML) [6] as the main way to convey informational representations of resources in order to serve them on the Web as shown in Figure 2.

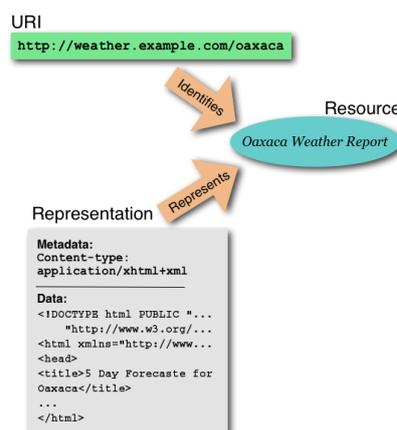


Figure 2: Relationships between identifier, resource, and representation [7]

For the web pages represented as HTML to be served to the user, the Web relies on the HyperText Transfer Protocol (HTTP) which defines a set of steps that governs the conversation between a Web client and a Web server to deliver the Web page into the client side as shown in Figure 3.

In an HTTP session, the client asks the server for the HTML representation of a resource. For this to take place, the client uses the identifier of the resource based on its URI. This URI is encapsulated in an HTTP request with an appropriate HTTP method, mainly the GET method.

The whole request/response session takes place over an Internet connection such as a TCP connection. The Web server which hosts the document replies to the GET request by an HTTP response which is a textual message containing, among other things, the HTML content.

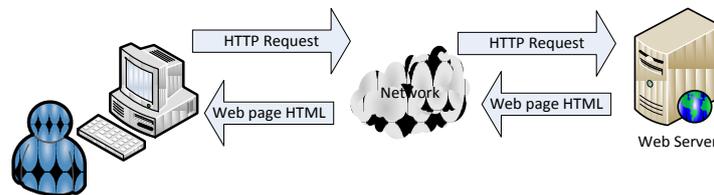


Figure 3: Basic HTTP architecture

HTML is a language for defining the layout of the content. The essence of the HTML Web content is that the text contains tags that govern the way it is displayed on the client side. Figure 4 for example shows a simple HTML document. The tag `<title></title>` tells the client application what the title of the window displaying the content should be. The tag `<h1></h1>` tells the client to display its content as a heading, possibly with a bigger and thicker font. While the tag `<p></p>` displays the content as a paragraph beneath the heading.

```
<!DOCTYPE html>
<html>
<head>
<title>Example Page Title</title>
</head>

<body>
<h1>A Heading</h1>
<p>A paragraph.</p>
</body>
</html>
```

Figure 4: An example HTML Web page

A client that can establish HTTP sessions and interpret HTML content and display it accordingly is a universal client as it works for all applications conforming to the Web standards. Such a client is called a browser with many available options including Firefox, Chrome, Safari, and Internet Explorer.

As can be seen in the above discussion, the Web is designed to convey textual content mainly to humans. It uses a set of standards to make this scalable, as well as a set of standards such as HTML, and CSS to display it to the end client in a way that the original author intended to be. The Web also employs hyperlinks to allow users to navigate through the content from one page to another. Nonetheless, the Web is not a friendly place for machines. A software agent cannot make sense out of the content. Thus, a scalable automated understanding of Web content is not possible with its original form.

3.2 The Semantic Web

In 2001 Burners Lee et al. addressed the problem of automated processing of Web content by software agents through a radical reconsideration of the Web stack of standards in a seminal paper in Scientific American [8].

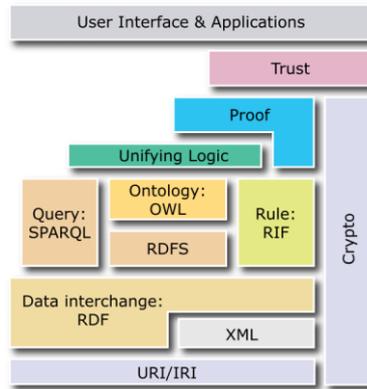


Figure 5: The Semantic Web layer cake diagram [9]

Figure 5 illustrates the main components of what the Semantic Web is conceived to be. One can still recognize URI as a main thing to identify resources on the Semantic Web. HTTP is also an important part as it can deliver any content and not necessarily HTML alone. Nonetheless, a drastic evolution can be seen in the new architecture which concerns the information representation of resources. In the Semantic Web resources are represented by structured data. The structured data follow a graph-based representational framework called the Resource Description Framework (RDF) [10].

In RDF, a piece of information is a triple of <subject, predicate, object>. Subjects, predicates and objects are typically URIs of resources. URIs can point at resources in the same domain or in other domains. The result is a big graph of interconnected data on the Semantic Web. Figure 6 shows an example RDF graph that describes the resource cygri to be of type Person, with the name Richard Cyganiak and the location of somewhere near Berlin. One can notice the use of prefixes to abbreviate URIs as in the original use of URIs in the Web.

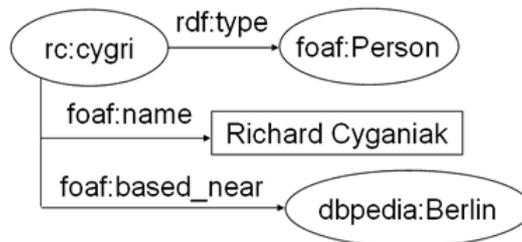


Figure 6: An example RDF graph [11]

RDF can be serialized in various formats, including XML and N3 as shown in Figure 7.

```
# RDF links taken from Tim Berners-Lee's FOAF profile
<http://www.w3.org/People/Berners-Lee/card#i>
  owl:sameAs <http://dbpedia.org/resource/Tim_Berners-Lee> ;
  foaf:knows <http://www.w3.org/People/Connolly/#me> .

# RDF links taken from Richard Cyganiaks's FOAF profile
<http://richard.cyganiak.de/foaf.rdf#cygri>
  foaf:knows <http://www.w3.org/People/Berners-Lee/card#i> ;
  foaf:topic_interest <http://dbpedia.org/resource/Semantic_Web> .
```

Figure 7: An example RDF N3 serialization [11]

To define the semantics of terms used in the RDF graph, ontology languages such as RDF schema [12] and OWL [13] are used. Ontologies are a way to explicitly define semantics where every concept or property has a URI. The language has constructs to specify relationships between the concepts such as `rdfs:subClassOf` which defines that a class A is of the kind B, another class.

A substantial part of the Semantic Web that has been actively developed is the Structured Query Language for RDF (SPARQL) [14] which is similar to SQL. In SPARQL users can specify queries composed mainly of graph patterns that can then be submitted to an RDF data store. The RDF data store matches the query against its internal data and return a set of results as specified. For example, Figure 8 shows an example SPARQL query that ask for resources of type Person with their acquaintances.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?anotherPerson
WHERE {
  ?person a foaf:Person.
  ?person foaf:name ?name.
  ?person foaf:knows ?anotherPerson.
}
```

Figure 8: An example SPARQL query

The original Semantic Web vision defines upper level layers in the stack including rules which add to the capacity and expressivity of ontology languages. Proofs, trust and other aspects are also part of the Semantic Web which inherits from Knowledge engineering communities rather than from the Web community. Thus, there has been a realisation that upper level layers of the stack are harder to tackle especially with less available data on the Web. The focus has shifted as a result to lower levels to publish data on the Web with the basic possible principles of the Web and Semantic Web standards, resulting in a new chapter of the Semantic Web called Linked Data.

3.3 Web of Data

Linked Data targets the exposure of the hidden data in servers behind the tiers of Web servers and making it available through the Web. That serves an important goal of integrating data silos all over the Web through Web standards. The key is that most of that data is already structured, and thus moving it to conform to the standards of Web and the Semantic Web shall be a relatively easier step. Berners Lee [3] summarizes the principles of the Linked Data in four steps:

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
4. Include links to other URIs. so that they can discover more things.

Extending the abstract view of the Web architecture, URI can still be used to identify entities or resources. Representations of such resources though are not extended with the RDF content. When a software agent (client) asks the Web server for a resource, it could ask for the RDF representation of that resource in what is called content negotiation, see Figure 9.

Once the RDF representation is returned to the client, it can parse it according to the serialization used and then make use of the links as well as the references to ontologies and vocabularies in order to make sense of the relationships between entities and make higher level reasoning.

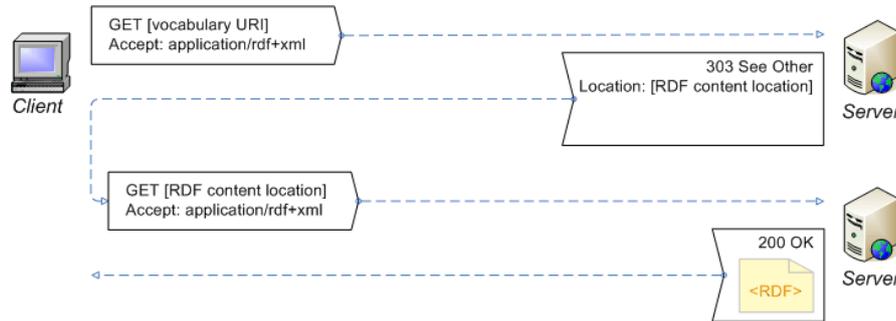


Figure 9: Content negotiation in Linked Data [11]

Linked Data has been seen as a cornerstone in the Open Data initiative and thus Berners Lee [3] defined a 5-stars scheme that can be used to publish Linked Data on the Web as shown in Figure 10.

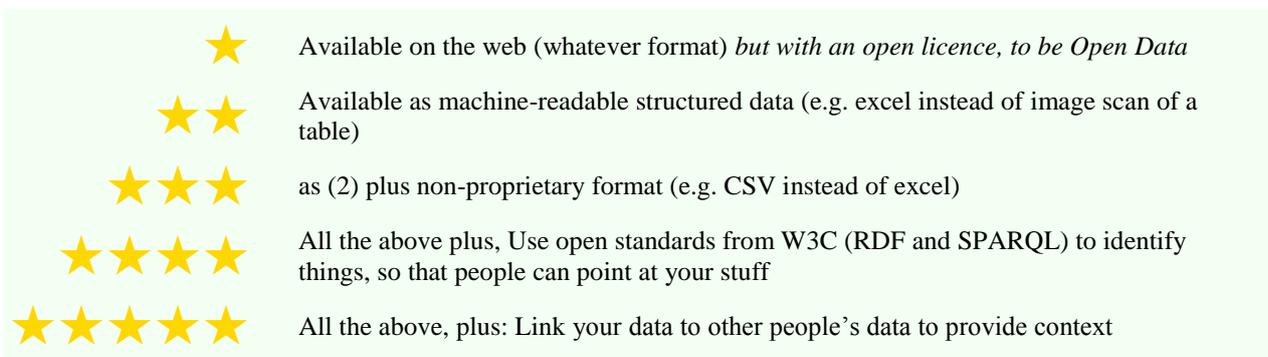


Figure 10: 5 stars scheme to publish Linked Open Data on the Web [3]

Many projects have been launched in response to the need for availing as much Linked Data as possible and realising the Semantic Web vision through the Linked Data principles. One notable project is the DBpedia [15] which is an online repository of structured content available in RDF and can be queried through SPARQL. DBpedia is extracted from Wikipedia and thus contains a large amount of broad and comprehensive structured information. If a resource is looked up in DBpedia, the content negotiation specifies if an HTML representation or an RDF representation is returned and then the content is sent back through HTTP response. Figure 11 shows the response to the request of a URI <http://dbpedia.org/resource/Water>.

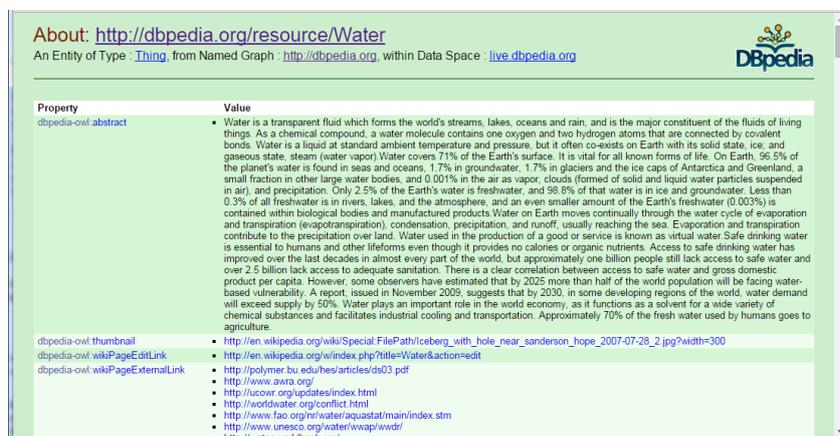


Figure 11: Dereferencing <http://dbpedia.org/resource/Water> [15]

Projects like DBpedia has emerged over time with resources being cross-used throughout all of them. As a result, a big cloud of inter-linked data stores has emerged as illustrated in Figure 12.

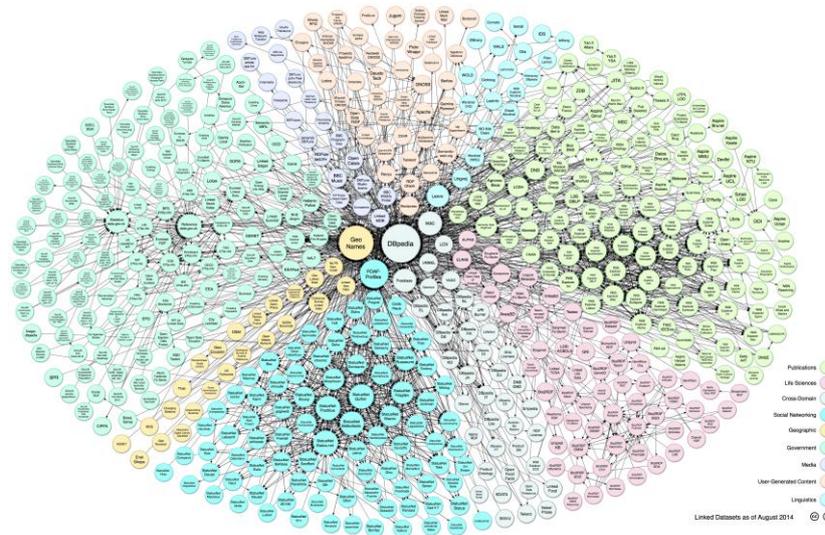


Figure 12: Linked Open Data cloud [16]

3.4 Linked Water Data

Based on the discussion above, drinking water supply spans multiple aspects of activities starting from extraction and production, through distribution and consumption, and ending with recycling. Thus, water data is likely to exist in many data silos which makes Linked Data a good candidate for tackling water data integration and allows a holistic view of all aspects related to water.

By projecting previous Linked Data principles to the water case, we could end up with main steps that shall guide the publication of Linked Water Data:

1. Use URIs as names for things: where thing here can refer to sensors monitoring water flow, locations where water is generated, spaces where water is consumed, people dealing with water lifecycle, taps, pipes, buildings, etc.
2. Use HTTP URIs so that people can look up those names: software clients shall use HTTP to negotiate content with water data servers. Such clients shall be able to establish HTTP sessions with servers. Clients can be user interfaces, dashboards, data analytics tools, stream processing engine, and collectors for data management systems, etc.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL), RDF shall be used to model water data, thus a proper ontological model for water needs to be defined using Web ontological languages to define concepts such as WaterFlow or CubicMeter, and predicates such Consumes or FlowThrough, etc. SPARQL can be used then by the software clients to query the resulting RDF data.
4. Include links to other URIs, so that they can discover more things: URIs used by different parties in a water lifecycle shall refer to each other, which would be the case for multiple departments in an airport, or multiple partners in the project. By doing this, a holistic view of water management can be achieved.

It is also important to reuse existing vocabularies that are proven to be effective and compliant with most of the sensor modelling standards. Indeed, we plan to use Semantic Sensor Network (SSN) ontology [17] for modelling sensors and their observations. In fact, SSN is built while being compliant to the W3C and the Open Geospatial Consortium (OGC) [18] standard SensorML [19].

3.5 Conclusion

We saw how Linked Data came into existence and how that can benefit the process of publishing Water data. In the following section we will review relevant technologies that together with linked data principles can systematically build a Linked Water Dataspace.

4 State of the Art Analysis

In the following we review important and relevant contributions in four main areas: data adapters, platforms for linked dataspace, entity management, and architectures for providing real-time decision support when sensor data is a part of the dataspace.

4.1 Data Adapters

There exist a number of tools and standards to transform structured data into RDF. For relational databases, R2RML [20] is a W3C Recommendation that describes a language for expressing customised mappings from relational databases to RDF datasets. R2RML mapping is itself written in RDF and provides powerful capabilities to transform relational databases into RDF data. D2RQ¹ and Ultrawrap [21] are two popular implementations of the R2RML standard. D2RQ is available for free and is an open source project available under the Apache License V2.0. Ultrawrap is a commercial product from Capsenta. As shown in Figure 13, Ultrawrap has two parts: compile and server. The compile component is responsible for the RDF mapping. The server component is responsible for managing SPARQL queries.

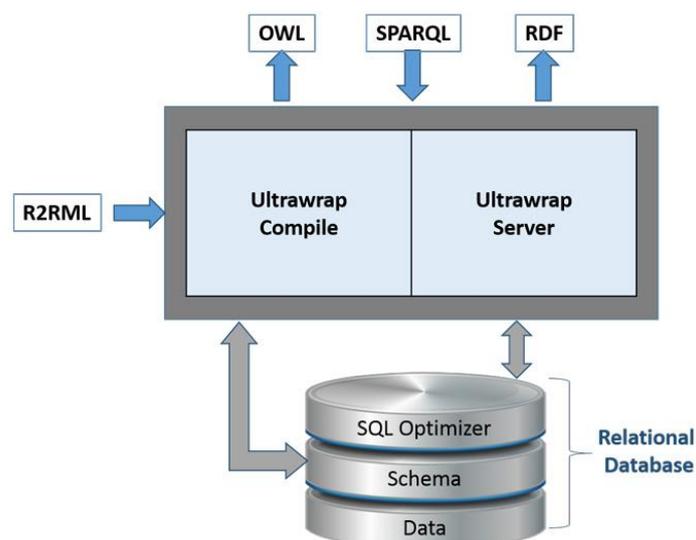


Figure 13: Ultrawrap architecture²

When it comes to general CSV and Excel data, there is no standard to describe the mapping. RML [22] defines an extension of R2RML to support transforming tabular data into RDF. RML abstracts items that are specific to relational databases in R2RML and proposes a more general specification that can support tabular data in general. The recently formed W3C CSV on the Web working group³ is working on a defining a standard to describe Mapping CSV into RDF. The final version of the standard is not released.

Basic conversion of tabular data into RDF produces a single resource per row. Properties of the produced resources are derived from the column headers and have cells contents as their literal values. All RDF resources generated have the same structure, a star-shaped graph, and custom graphs cannot be produced. Convert2RDF [23] is an example of tools supporting this type of naive conversion.

¹<http://d2rq.org/>

²<http://capsenta.com/architecture/>

³http://www.w3.org/2013/csvw/wiki/Main_Page

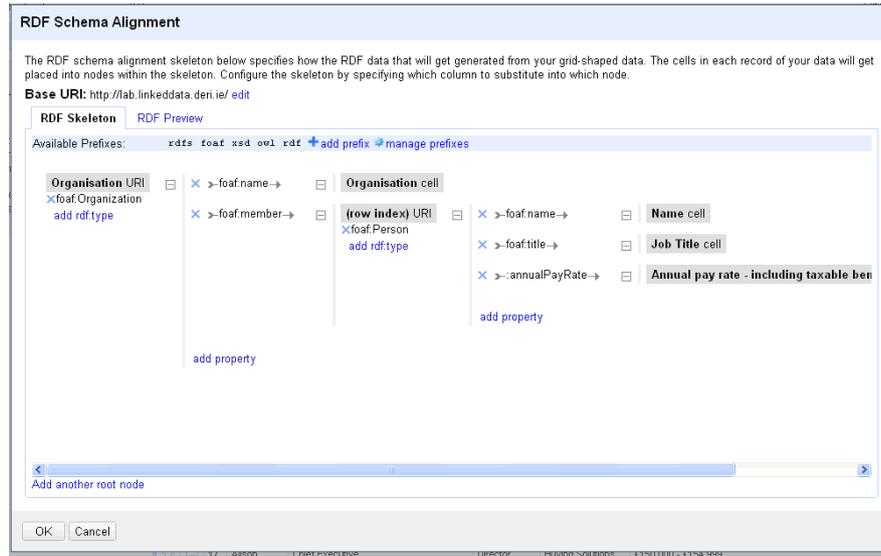


Figure 14: Google Refine RDF extension user interface 1

The approach described in [24] is to get basic RDF data by automatic conversion and then gradually improve its quality. Quality improvement is achieved by constantly applying a selection of predefined “enhancement operations”. Examples of the enhancement operations are converting literals into resources and defining *rdfs:subClassOf* or *rdfs:subPropertyOf* relations. Applying enhancement operations, while enabling exporting rich RDF data, results in new versions of the data and adds the overhead of managing multiple versions.

RDF123 [25] and XLWrap[26] support rich conversion and full control over the shape of the produced RDF data. Both of them use a template graph to describe the structure of the intended RDF data with predefined expressions to enable the use of cells contents in the result. XLWrap does not only support tabular data, but also more complex data layout such as multiple-tables and multiple-sheets data.

RDF Extension of Google Refine¹ provides a graphical user interface (GUI) that allows easy definition of mappings into RDF. The RDF Extension of Google Refine builds on top of the powerful capabilities of Google Refine to support cleansing as well as transforming tabular data. It is available as open source under BSD License. Figure 14 shows an example of RDF Extension main interface.

Grafter² is a product developed by Swirrl. Grafter defines a Domain Specific Language (DSL) which allows the specification of transformation pipelines that convert tabular data into linked data formats. Grafter can support transformation into RDF in a streaming fashion to allow processing large amounts of data. Grafter is an open source project available under Eclipse Public License. Figure 15 shows an example of mapping using Grafter DSL.

¹<http://refine.deri.ie>

²<http://grafter.org>

Tabular Input & Transformations

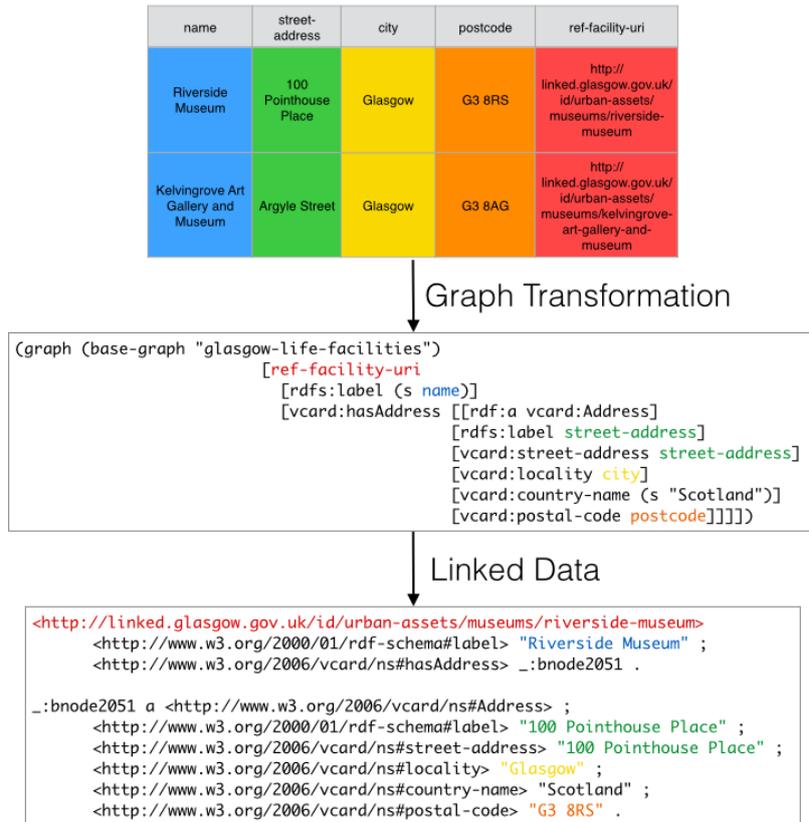


Figure 15: mapping tabular data to RDF using Grafter DSL2

TARQL¹ (Transformation SPARQL) is a SPARQL-based data mapping language. It uses SPARQL syntax to transform CSV / TSV data into RDF. The transformation in TARQL is expressed as a CONSTRUCT SPARQL query (see Figure 16 and Figure 17).

```
1 CONSTRUCT { ... }
2 FROM <my_file.csv>
3 WHERE {
4   ...
5 }
```

Figure 16: Sample SPARQL query in TARQL

```
1 CONSTRUCT { ... }
2 WHERE {
3   VALUES (...) { ...}
4   ...
5 }
```

Figure 17: Equivalent SPARQL query

Input data is used as a table of bindings. It means, that the data itself can be modified using the SPARQL syntax (the tool supports SPARQL 1.1 syntax). The conversion process has multiple configuration options and supports the conversion of large files through the use of streaming. TARQL is available as a command-line tool. It's written in Java and it is based on Apache ARQ² and released as an open source project available under BSD License.

¹<https://github.com/cygri/tarql>

²<http://jena.apache.org/documentation/query/>

4.2 Linked Data Platforms

Linked Data Platform is a set of patterns for building RESTful HTTP services that describes their resources as Linked Data. As shown in Figure 18, a resource may be fully represented in RDF (Linked Data Platform RDF Source (LDP-RS)) or have only an RDF representation (Linked Data Platform Non-RDF Source (LDP-NR)). While being managed by an LDP, each resource is referred as a Linked Data Platform Resource (LDPR).

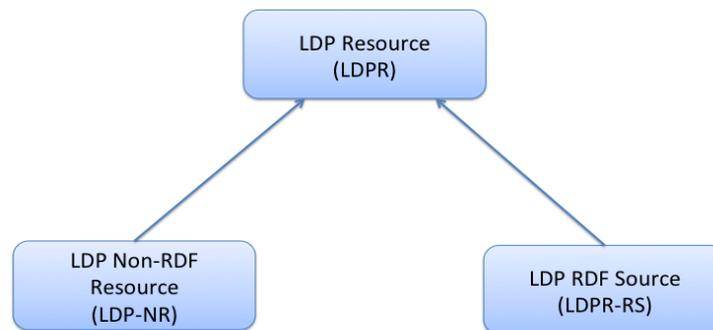


Figure 18: Linked Data Platform types

W3C provides a recommendation for the architecture of a Linked Data Platform 1.0¹ (LDP). LDP defines a set of rules for HTTP read-write operations (accessing, updating, creating and deleting from the server) on web servers.

Most of the Linked Data Platform Resources (LDPRs) are domain-specific and contain data for an entity in some domain (i.e. governmental or scientific). To group the collection of linked documents or information resources LDP defines the Linked Data Platform Container (LDPC). This idea is very beneficial for applications model that use collections of resources. In the following, we describe existing projects that support the implementation of LDP.

4.2.1 Virtuoso

Virtuoso Universal Server² is a multi-model data system. Single universal server combines the functionality of file server, virtual database, relational database management system and object-relational database as well as the RDF graphs data management.

As shown in Figure 19, Virtuoso has a build-in support of LDP functionalities. It can be used as LDP Client and as a LDP Server. Virtuoso LDP Client is generating the HTTP requests and processing HTTP responses according to the defined rules. Virtuoso LDP Server is processing the HTTP requests and generating the HTTP responses according to the defined rules.

Virtuoso serves RDF data via a Linked Data interface and a SPARQL endpoint. The data can be stored in Virtuoso data store or can be created on the fly from relational databases based on a predefined mapping. Virtuoso Universal Server has proprietary license but there is an Open Source version, which is available with GPLv2 license.

¹ <http://www.w3.org/TR/ldp/>

² <http://virtuoso.openlinksw.com/>

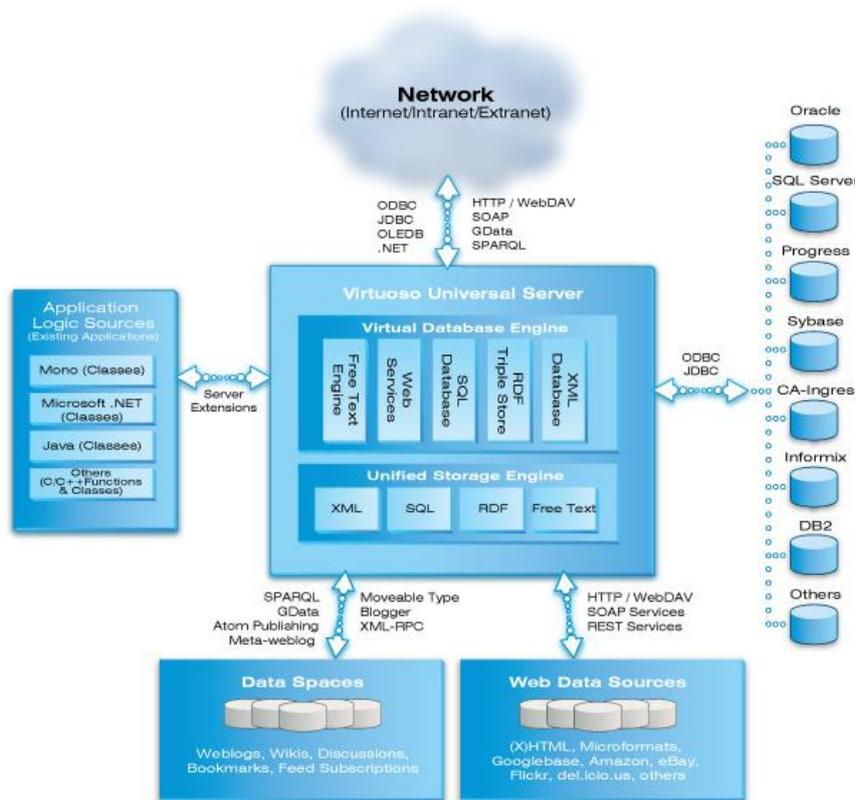


Figure 19: Virtuoso Architecture Diagram2

4.2.2 LinDA

LinDA is an acronym for LINKed Data Analysis [1]. General idea of the project is to provide a central platform that allows (Linked Open Data) LOD community to analyse the LOD Cloud. First implementation includes four tools for Linked Data analysis:

- VOID description generator for the RDF dataset,
- formal concept analysis tool,
- schema index generator,
- schema information analysis tool.

LinDA is a data driven platform. All results and datasets are cached in the system. LinDA web interface provides information about all uploaded datasets and overview about available services and computed structure analysis.

4.2.3 Pubby

Pubby¹ is a Linked Data Frontend for SPARQL Endpoints. It is implemented as a Java web application. The main idea of the project is to provide a Linked Data Interface for the endpoints, which are accessible only by SPARQL client applications that use the SPARQL protocol.

¹ <http://wifo5-03.informatik.uni-mannheim.de/pubby/>

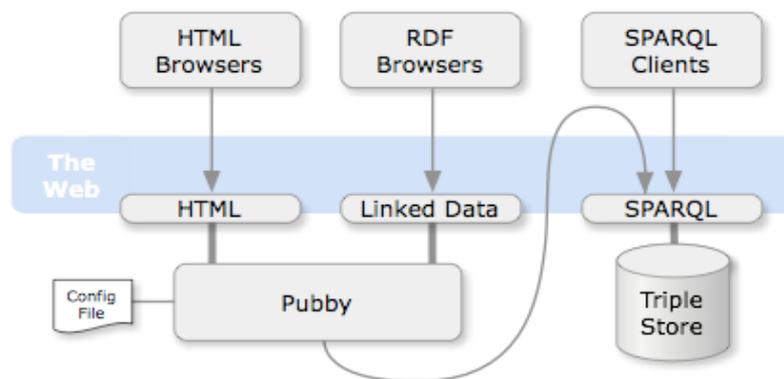


Figure 20: Pubby Architecture Diagram 1

Pubby configuration uses Turtle syntax. It consists of a mapping that translates endpoint resources URI into URIs handled by Pubby. The tool is requesting the mapped URI at the SPARQL endpoint, converts the URI and passing back the response to the client. An overview of Pubby Architecture is depicted in Figure 20.

There are few limitations of Pubby. For example, it requires SPARQL endpoint to answer DESCRIBE queries, multiple datasets may not work as expected as well as it does not support hash URIs. Last official release was in 2011 as Pubby version 0.3.3, but the code base is constantly updated. Pubby is available as an Open Source project licensed with Apache License, Version 2.0.

4.2.4 Apache Marmotta

Apache Marmotta¹ is an Open Platform for Linked Data. It is implemented as Service-Oriented Architecture and it consist of a collection of modules. One of the modules is the LDP. Its goal is to provide and Open Source implementation of an LDP. Current Apache Marmotta version (3.3) is compliant with Linked Data Platform 1.0 but it is not yet complete, which means that there are some implementation restrictions.

The main features of the platform are:

- Read-Write Linked Data
- RDF triple store
- SPARQL endpoint
- Data caching

Marmotta is one of the most active projects of the Apache Software Foundation². The project is available under the Apache 2.0 license.

4.2.5 LDP4j

LDP4j³ is a java-based Open Source framework for the development of read-write Linked Data applications. The framework provides the components useful during the development of LDP-Client and LDP-Server applications. It handles LDP-communication letting the developers focus on the application-specific business logic. Moreover LDP4j provides a set of middleware

¹ <http://marmotta.apache.org/>

² https://blogs.apache.org/foundation/entry/the_asf_asks_have_you2

³ <http://www.ldp4j.org>

services for the development of read-write Linked Data applications. LDP4j is available under the Apache 2.0 license.

4.2.6 OpenCube Toolkit

OpenCube¹ is the acronym of the project founded by the EU Seventh Framework Programme. The full name of the project is: Publishing and Enriching Linked Open Statistical Data for the Development of Data Analytics and Enhanced Visualization Services. The project aims at developing software tools that facilitate publishing of high-quality Linked Statistical Data and reusing distributed Linked Statistical Data in data analytics and visualizations. OpenCube is an ongoing project, and the final results are expected by the end of the year 2015.

The beta version of the OpenCube Toolkit²(see Figure 21 for a sample page) is already available at the project website. The software is built upon the Open Source version of Information Workbench³-Community Edition platform. The platform was used to ensure generic low-level functionalities such as shared data access, logging and monitoring. OpenCube Toolkit is still in development.



Figure 21: OpenCube Toolkit sample page

OpenCube Toolkit functionalities can be categorised as follows:

- **Data Publishing:** The platform allows data conversion to RDF from multiple formats: legacy tabular data (such as CSV/TSV files), relational databases, XML files and more. Moreover the data can be imported from external portals, such as CKAN instance (Open Source data management system). RDF data can be uploaded directly to the system or imported from selected a URL or a SPARQL endpoint.
- **Data Reusing:** The idea behind the project is to develop the tools that will allow easy data integration, analysis and visualisation. OpenCube Toolkit user interface is based on the templates system and widgets. A widget is an interface element which can be configured and embedded into a wiki page. For example it can display a list of the corresponding semantic resources.

¹ <http://opencube-project.eu/>

² <http://opencube-toolkit.eu/>

³ http://www.fluidops.com/en/portfolio/information_workbench/

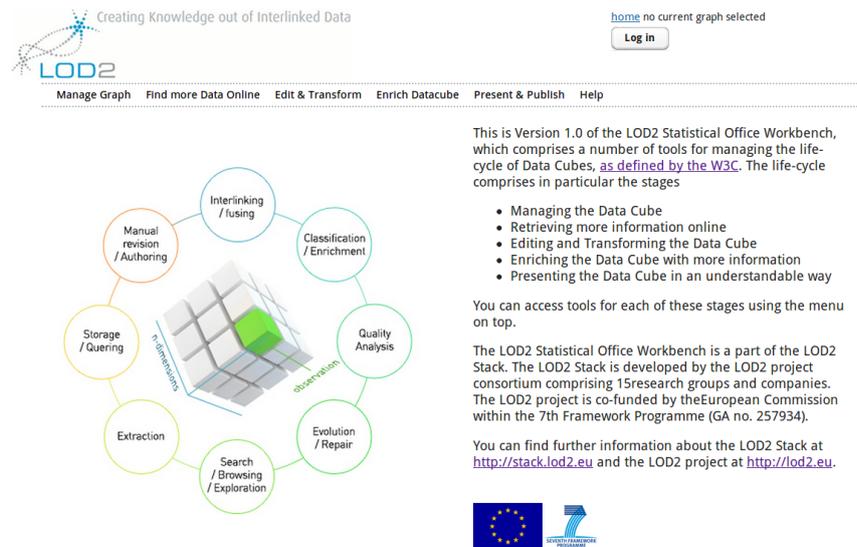


Figure 22: Statistical Workbench dashboard

4.2.7 LOD2 Linked Data Stack

The LOD2 Linked Data Stack¹ is a large-scale project founded by the EU Seventh Framework Programme. The project aims at providing variety of tools for Linked Data publication and visualization. A unified environment and ease of use shall improve the quality and the coherence of the published data. Moreover the expected impact is the lower entrance barrier for data publishers and users and high performance of the RDF data management. Above all, it establishes trust on the Linked Data Web.

LOD2 Linked Data Stack defines the following steps for Linked Data life cycle:

- Storage
- Authoring
- Interlinking
- Classification
- Quality
- Evolution / Repair
- Search/Browsing/Exploration

Based on the need of the statistical domain LOD2 Statistical Workbench (see dashboard on Figure 22) was released. It is a specific configuration of the LOD2 Linked Data Stack. The platform provides the interface that allows users to find the information, transform it to RDF, refine it and visualize after the analysis is made. The acquired data can also be published.

The platform is available as a collection of Debian packages and as pre-installed virtual machine images. The LOD2 Linked Data Stack targets Ubuntu 12.04 LTS as operating system.

¹ <http://lod2.eu/>

4.2.8 CKAN

CKAN¹ is the Open Source data portal software (see dashboard in Figure 23). CKAN provides a set of tools for data sharing, publishing and finding. Moreover it provides a rich RESTful JSON API for querying and accessing dataset information.

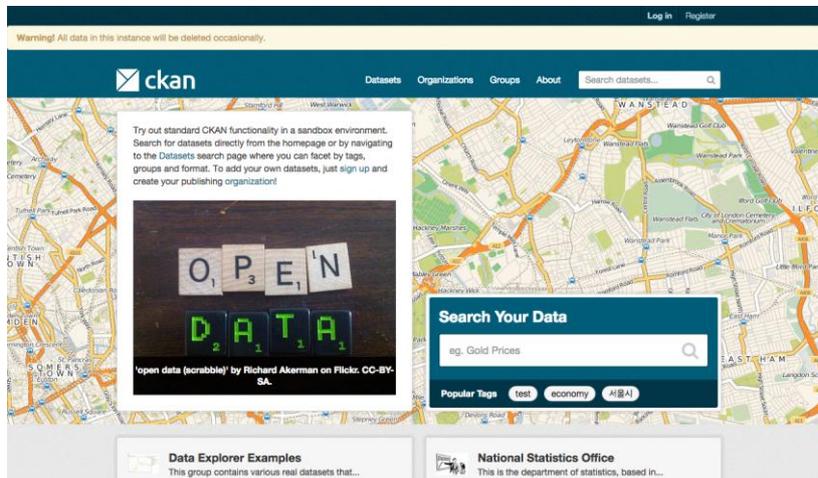


Figure 23:CKAN dashboard

The main functionalities of the platform are:

- Data Storage
 - Raw Data
 - Metadata
 - Data versioning
 - Data Access Restrictions (public / private)
- Data catalog
 - Searching
 - Tagging
- Visualization
 - Table View
 - Graphing
 - Mapping
 - Imaging
 - Custom solutions / extensions
- Theming & CMS integration

CKAN is released as Open Source software.

4.3 Entity Management

Information about the core entities within water management processes is critical to support accurate analysis and decision making. In the following we summarize different entity management approaches.

¹ <http://ckan.org/>

4.3.1 Master Data Management

Master Data Management (MDM) [27]–[30] is a process oriented approach that involves both technical implementation and organizational support for creating a definitive source of reference data about enterprise entities. MDM involves implementation of strict control over business processes through policies defined for data creation, update and removal. This control is achieved under guidance of an MDM council which includes members from senior management, business managers and data stewards [31]. The MDM council not only defines procedures for data acquisition but also defines precise business rules to gauge and improve quality of entity data.

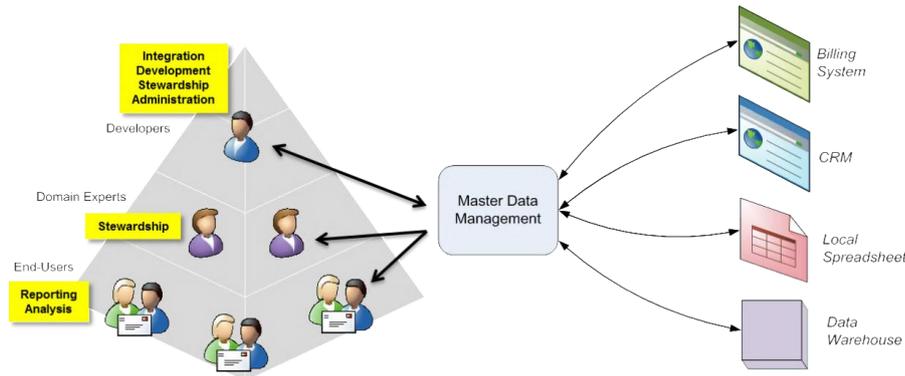


Figure 24: Illustration of master data management based solution for entity data management

Recently master data management has become a buzz world in enterprise software industry because of its potential benefits, if successfully implemented. However, this approach is heavily centralized and labour intensive, where the cost and effort in terms of expertise can become very high as depicted in Figure 24. MDM councils are responsible for inclusion or exclusion of sources from integration efforts as well as definition of business rules for guaranteeing certain levels of data quality, in *top-down* manner. Personnel within organization are given stewardship roles to run frequent quality checks. The end result of all these upfront efforts is an authoritative source of enterprise master data with single version of truth about entities that is utilized by end users in their daily reporting and analysis tasks.

The main benefit of successful MDM implementation is readily available high quality data about enterprise entities, in effect reducing the work of business analysts. Standardization of information development across lines of business is a secondary benefit. Although the notion of MDM has been very attractive for enterprise the actual success rate for these projects has fallen short of expectations. Recent studies estimate that more than 80% data integration projects in enterprises either fail or overrun their budget [32], [33]. The significant upfront costs in terms of development efforts and organization changes make the undertaking difficult to achieve across large enterprises. Concentration of data management and stewardship among a few highly skilled individuals, like developers and data experts, also proves to be a bottleneck. Due to limited number of skilled human resources, small percentage of enterprise data comes under management. Subsequently the scalability of MDM becomes major issue when new sources of information are added over time. Furthermore as the cost of data storage is decreasing by day, the amount of data relevant to enterprise and its entities is also increasing exponentially [32], [34]. Enterprises are unable to cope with the scale of data generated within their boundaries. Similarly, as the *Web of Data* becomes important and as enterprises collaborate more there will be need for enterprises to manage relevant data existing outside their boundaries, as illustrated in Figure 25. Moreover external data sources from business partners further exasperate this problem.

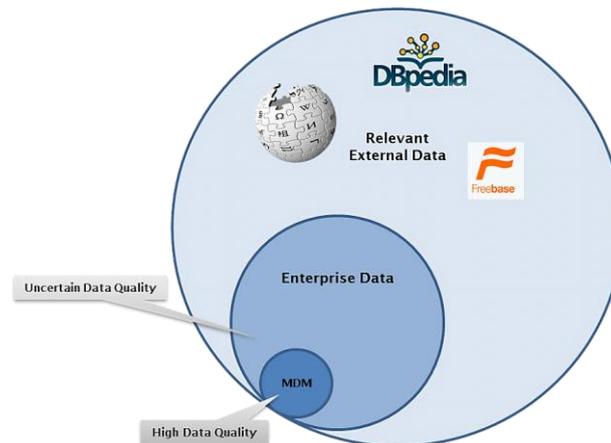


Figure 25: Relative volumes of data relevant to an enterprise, where MDM manages small part of data available in internal and external sources

4.3.2 Crowdsourced Entity Management

Web 2.0 has changed the role of humans on internet from information consumers to information contributors. Owing to this change there has been an explosion in number of systems gathering people most suitable to solve a problem in open calls. In this direction, *crowdsourcing* [35] techniques has been applied to recommendation systems [36], question answering [37]–[39], and knowledge creation [40], [41]. For example Wikipedia¹ follows a *document-centric* approach for collaborative knowledge creation and management [40]. Likewise there have been efforts to develop *entity-centric* repositories of authoritative structured information for web-scale use such as Freebase² [42], UniPort³ [43] and OKKAM⁴[44]. In this regard, the Freebase and OKKAM are targeted specifically towards the creation of authoritative knowledge bases of mostly structured entities. Both approaches rely on contributions from community of web-users for long term success.

Freebase is web based entity store of mostly structured data initially created by extracting information from Wikipedia and other sources [42], [45]. Users are allowed to create their own entity types and their entities in their personal space. Freebase can be considered as large curated database with strict administrative controls over public entities. It is supported by a question answering service called RABJ [46]. The objective of this service is to support data processing activities with human supervision by allowing user to distribute specific data quality questions to community of end-users.

OKKAM project focuses on creating a web scale entity naming system (ENS) for linked data [47], which is similar in design to domain name system responsible for web address management. Since RDF data representation relies on URIs for identifying entities, OKKAM uses set of minimum descriptors of entities to maintain uniqueness. A canonical URI is created for each entity along with set of all URIs describing the same entity in different sources. The entity linkage based on *name-feature-matching* algorithm are central to successful quality management [48] in OKKAM. Furthermore web users are allowed to subscribe to entities of their interest in turn monitoring changes and contributing in data quality efforts.

¹ <http://www.wikipeda.com>

² <http://www.freebase.com>

³ <http://www.uniprot.org>

⁴ <http://www.okkam.org>

Moving away from the public domain the *Cimple*¹ project concentrates on information management for specialized topics for example database researchers or bio-informatics [49]. As a proof of concept a prototype (*DBLife*²) was created following a top-down, compositional and incremental approach [50]. Developers initially design the system by specifying entities to be extracted and integrated from unstructured sources following operators based design approach. Then the system automatically expands the coverage based on mention of other sources in data while managing quality with help of end users.

4.3.3 Collaborative Entity Management

Considering the limitations of master data management and large-scale crowdsourcing, a hybrid approach for management of entities was proposed [51]–[53], as shown in Figure 26.

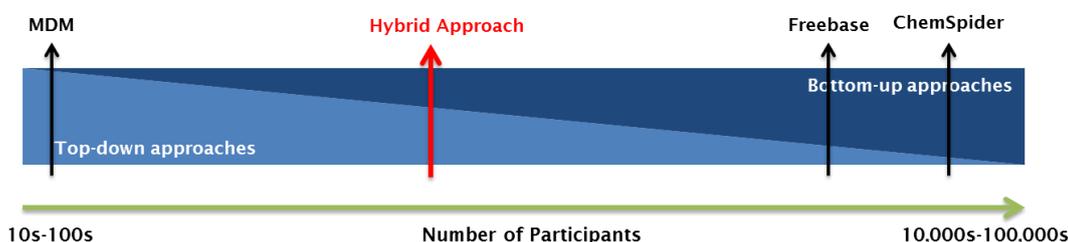


Figure 26: In continuum of entity management approaches the number of contributors increase from top-down to bottom-up. A hybrid approach would help expand boundaries of enterprise entity management

The hybrid approach utilizes community participation to increase the quality data lying outside of existing MDM efforts. In the proposed hybrid setting, *experts* (e.g. developers, data stewards, and data analyst) will be able to specify data integration and quality requirements of entity consolidation, with the objective of outsourcing recurring tasks to non-experts, within or, outside the enterprise. *Non-experts* are defined as the people with limited technical knowledge of data management and integration; however they can be knowledgeable within their particular domain. Figure 27 highlights the distribution of integration and stewardship efforts from experts to non-experts which differentiates proposed approach from existing expert dependent approaches.

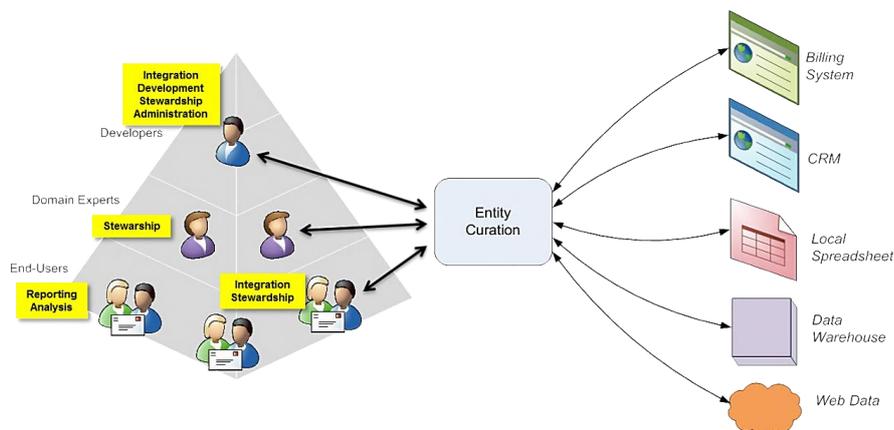


Figure 27: Community collaboration based approach for management of entity data

Web data management techniques are generally designed to perform better in terms of volume of data and number of users. On the other hand master data management technologies underline the

¹ <http://pages.cs.wisc.edu/~anhai/projects/cimple/>

² <http://dbliflife.cs.wisc.edu/>

significance of tight control over data acquisition, integration and management process, due to criticality of business decision making as well as audit requirements. CAMEE aims to exploit the strengths of collaborative entity management, combined with top-down control of master data quality to expand the boundaries of data under management [51], [54], [55]. CAMEE builds on the concept of human computation to formalize explicit feedback, for data quality and uncertainty management tasks, within entity consolidation.

4.4 Lambda Architecture

The Lambda architecture was proposed with the aim of allowing seamless ingestion and processing of streaming events data. The stream of events could be sourced from varieties of systems such as sensors, database logs, website logs, etc. In the following, we discuss two popular reference implementations of the Lambda architecture.

4.4.1 DRUID Platform

The DRUID [56] is a data store designed to be deployed as the serving layer for the implementation of Lambda architecture. The primary design objective of the data store was to ingest large-scale streaming data and enable queries over it in real-time. The query processing is seamless between real-time data and historical data.

Figure 28 shows the high level overview of the various components in the Druid. There are four types of processing nodes in the Druid cluster, as described below:

- Real-time nodes are mainly concerned with ingest streaming events data. The real-time nodes aim to provide real-time data for immediate query processing. They also submit data chunks for long-term storage.
- Historical nodes maintain the historical data gathered over time by the real-time nodes. Additionally these nodes allow for the ingestion of batch data as a bulk load. Essentially these nodes are responsible to supporting the queries spanning over multiple dimensions and time periods.
- Coordination nodes are responsible for managing the data in historical nodes. These nodes decide when to load new data, how to partition data, when to discard old data, etc.

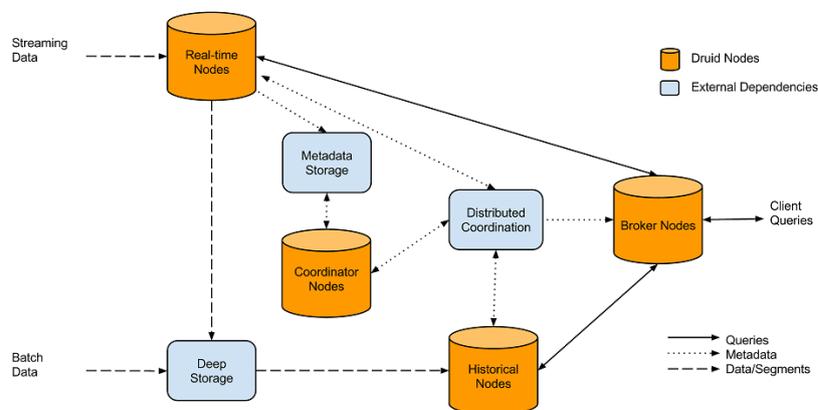


Figure 28: An architectural overview of the DRUID [56]

Broker nodes manage the query processing functionality but sending the data request to appropriate real-time and historical nodes. These nodes use the metadata information to location the required data and perform merge when receiving results from different nodes.

4.4.2 Apache Spark

Spark has initially been introduced as a framework for distributed parallel programming in tasks that has requirements different from common MapReduce applications. Mainly, analytics and machine learning jobs that needs iterative reuse of a working set of data objects [43]. SPARK achieves this by an abstraction called resilient distributed datasets (RDDs) which are distributed read-only collection of objects that are recoverable.

Apache Spark is an open source engine for data processing on large-scale [44]. Spark is designed to beat popular MapReduce frameworks such as Hadoop especially for iterative jobs such as machine learning and interactive analytics [43]. It is also important to note that Apache Spark is not meant to be a replacement technology for Apache Hadoop stack, it is rather proposed for fulfilling a different use case for real-time processing. Figure 29 illustrates the main components and libraries that form the overall family of apache Spark.

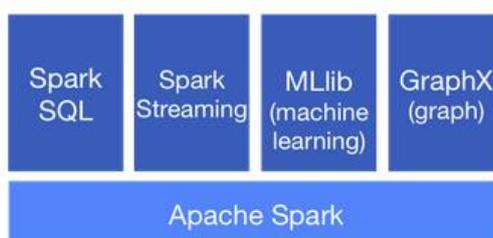


Figure 29: The Apache Spark family [44].

Spark SQL allows the querying of Spark RDDs using standard SQL language. It provides interfaces in multiple programming languages such as Java and Scala. That allows the execution of SQL queries as well as complex algorithms for data analytics. Spark SQL thanks to the Schema RDDs allows standard access to multiple formats such as relational data, Hive data, and Json files as shown in Figure 30. It also allows full integration with Hive so they can be plugged into the Spark data warehouse as shown in Figure 31. Spark SQL provides also seamless integration with existing Business Intelligence tools through standard data access connectivity such as JDBC and ODBC as shown in Figure 32.

```
sqlCtx.jsonFile("s3n://...")
    .registerAsTable("json")
schema_rdd = sqlCtx.sql("""
SELECT *
FROM hiveTable
JOIN json ... """)
```

Figure 30: Querying different data sources with Spark SQL [44].

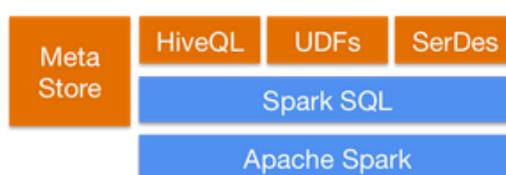


Figure 31: Integrating Spark SQL with Hive [44].



Figure 32: Standard data connectivity [44].

MLlib is a scalable machine learning library in Spark allowing the seamless execution of machine learning and data analytics algorithms over a Spark data warehouse. MLlib 1.1 contains algorithms such as linear SVM and logistic regression, classification and regression tree, k-means clustering, etc. It is usable within Java, Scala and Python and also executable over Hadoop data sources.

GraphX is the Spark library for unifying ETL, data analysis and graph data processing. It allows the manipulation of graph data as graphs of collections as well as transformation, joining, and writing custom graph algorithms using the Pregel API. GraphX is a good candidate for manipulating Linked Data in the Waternomics dataspace.

Spark streaming is the Spark library to manipulate, query, and transform on-the-fly data which is important for the speed layer in the Lambda architecture adopted for the Waternomics Linked Dataspace. Spark streaming allows the developer to write stream applications in the same way of writing batch applications which allows code re-usability. It provides high level operators as well as window operators. Spark streaming is fault tolerant and follows an exact once semantics. Spark stream operates over DStreams which are streams of DDRs as shown in Figure 33.



Figure 33: Standard data connectivity [44].

Spark can be deployed as standalone or in a cluster deployment mode as shown in Figure 34.

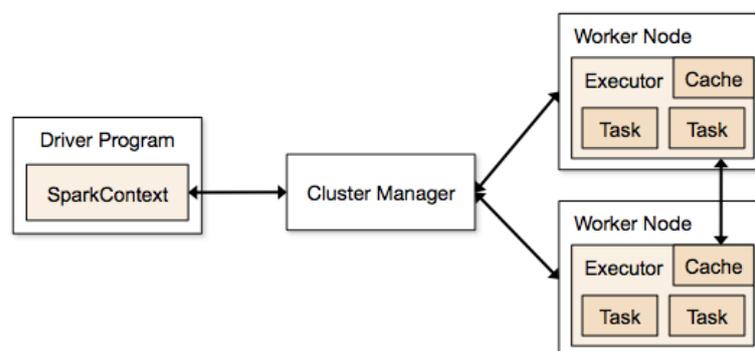


Figure 34: Spark cluster deployment [44].

4.5 Conclusion

Throughout the review of related work above, we could see different technologies that can be used to implement the adapters for Linked Dataspaces. A data catalog is one of the primary requirements of realizing a dataspace; therefore, we propose to use CKAN for this purpose. The catalog serves as a registry of data source, schemas, and entities. As summary of proposed

solutions for the remaining functional requirements, of a Linked Dataspace, is provided in Table 1: Analysis of the State of the Art.

Table 1: Analysis of the State of the Art

Functional Requirements	Challenge
Standardisation	Critical water entities need to be standardized and mapped their identifiers in different data sources.
Consuming Open Data	Various open source data can have various types/formats.
Publishing Linked Data	Publishes entities, schemas, open data, and aggregate data according may not be known to interested applications.
Data Linking	Mappings between entities and schemas with RDF and OWL relationships can be implicit and difficult to manage and maintain.
Real-time data / events	Events need to be consumed in a reliable, decoupled mode, with low latency.
Real-time Analytics	The need to serve real-time and historical data in aggregated form to applications.
Data integration	Full data integration is time consuming and forms an overhead that delays putting the applications into use.
Heterogeneity of Sensor Data Events	Events are semantically heterogeneous and agreeing on universal schema to support exact event processing is not scalable.
Enrichment of Sensor Data Events with Open Data	Sensor data events could lack data and need to be complemented dynamically with open data rather than dedicating ad-hoc enrichment logic by event engines.

5 A Realtime Linked Water Dataspace

Deliverable D1.3-Section 3 discusses an overall system architecture that contains three main layers: the hardware, the data, and the software, as shown in Figure 35.

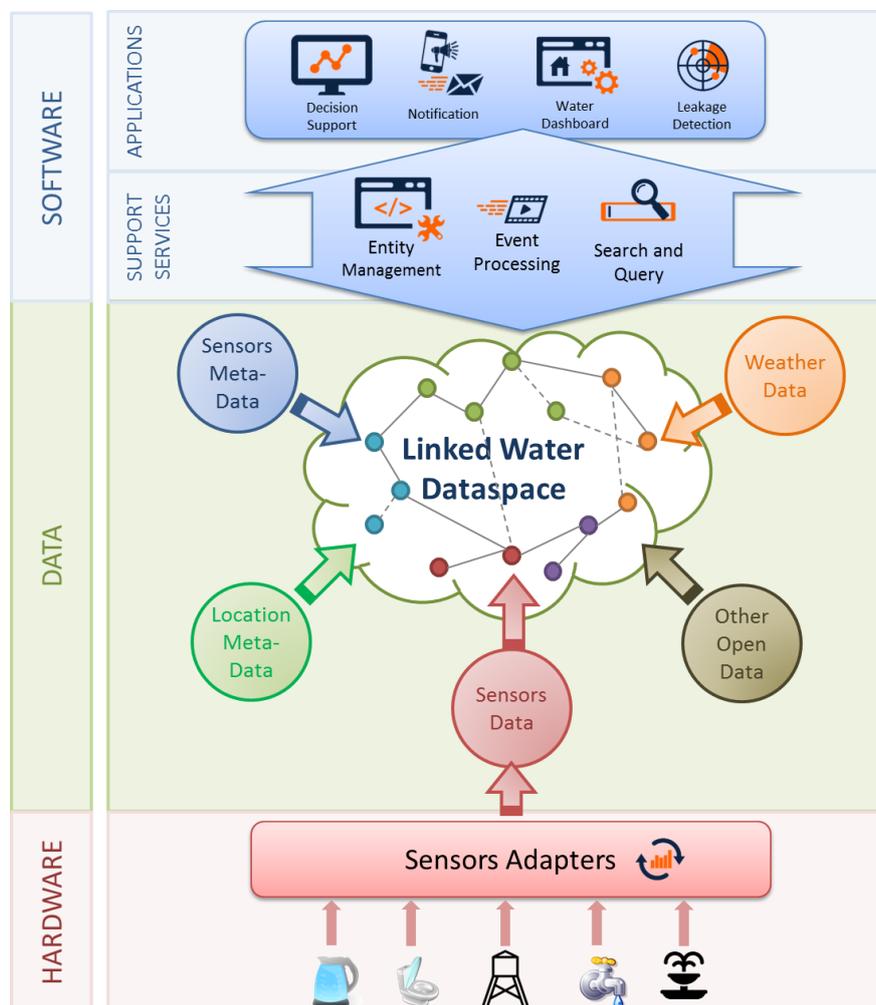


Figure 35: System Architecture.

In this section, we dig deeper into the data layer which is represented by the Linked Water Dataspace. By reviewing the current state of the art, we propose the following approaches for handling the requirements in the Linked Dataspace as shown in Table 2.

Table 2: Proposed Approaches for the Dataspace

Requirement	Approach
Standardisation	Standard compliant ontologies such as SSN will be used for describing sensors and their data. An entity management service that standardizes critical entities and maps their identifiers in different data sources. We propose to extend CKAN data catalog for this purpose.
Consuming Open Data	One of the support services that will be included in the dataspace consists of crawling weather data and integrating it in prediction services. Various open source data can have various types/formats, and an app might not know which open dataset is relevant to a particular task. Register each source in

	CKAN data catalog as an explicit indication of joining the dataspace. We propose to provide custom adaptors for each open data sources to make them available in dataspace.
Publishing Linked Data	The produced data by every support service will be published in RDF using linked data principles. Publish entities, schemas, open data, and aggregate data according Linked data principles, in RDF format. We propose to register each RDF source in CKAN data catalog as an explicit indication of joining the dataspace.
Data Linking	Relevant open data sources will be integrated in the dataspace through explicit links between entities. Add mappings between entities and schemas with RDF and OWL relationships. We propose to maintain high level mappings in CKAN data catalog and low level mappings maintained by each source.
Real-time data / events	The dataspace is design with respect to the lambda architecture that covers real-time event processing. We propose a complex event processing engine for managing live sensor events. Deploy a scalable message oriented middleware for passing data between real-time sources and applications. We propose to use middleware based on published/subscribe pattern, such as Apache Kafka.
Real-time Analytics	Real-time analytics of data is considered in the platform as part of the speed layer of the lambda architecture. Seamlessly serve real-time and historical data in aggregated form to applications. We propose to implement the serving layer of Lambda architecture using Metamarkets DRUID cluster.
Data integration	Lambda architecture in essence was proposed to deal with both historical and real-time data. This data integration is insured by relevant support services. Follow a pay-as-you-go paradigm and ingest, integrate, and aggregate real-time data. We propose to implement the speed and batch layers of Lambda architecture by employing Apache Spark for ingestion and aggregation of both real-time and historical data.
Heterogeneity of Sensor Data Events	The platform is designed to handle a variety of sensor types. Consequently for each sensor type, a dedicated collector is designed for collecting data and converts it into RDF for further processing. Use an approximate semantic matching model to process sensor events.
Enrichment of Sensor Data Events with Open Data	Sensor readings are further enriched by dedicated support services using open data. Dynamically enrich sensor events with open data sources to provide context to the data.

5.1 Dataspace Overview

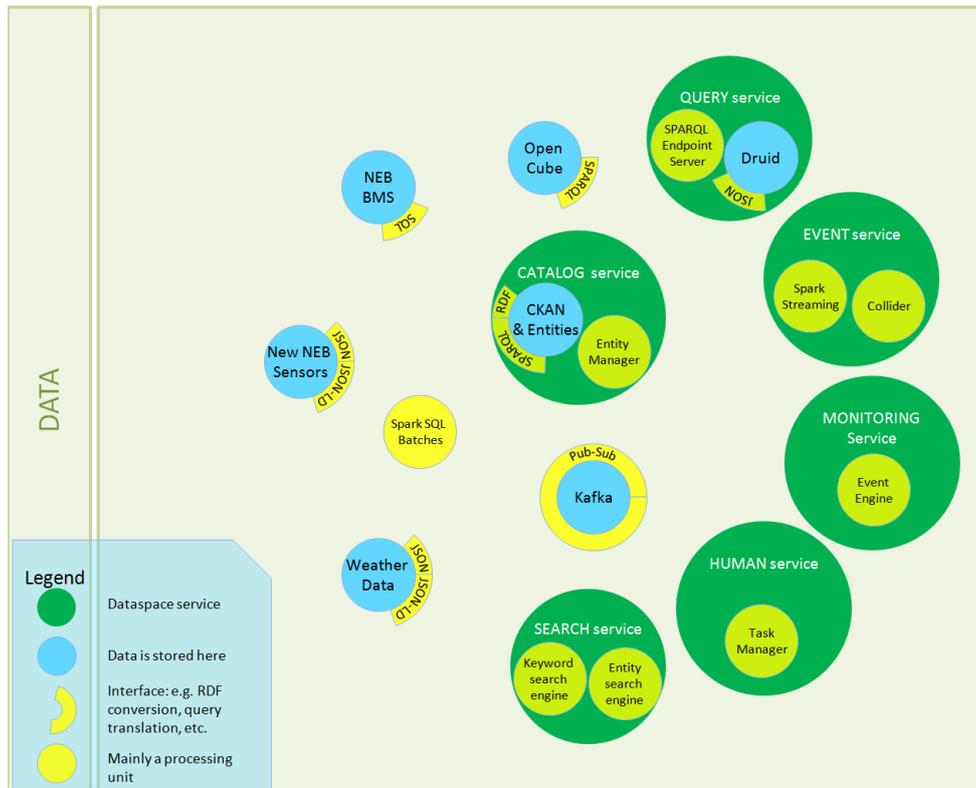


Figure 36: Linked Water Dataspace Overview- Components View

The Linked Water Dataspace shown in Figure 36 is a collection of water datasets along with a set of services that supports the dataspace. The dataspace is designed to be an incremental view of how water datasets join the computational space targeted by applications. In contrast to the classical one-time integration of datasets that causes a significant overhead, the Linked Water Dataspace adopts a pay-as-you-go paradigm. Water datasets join the space in an incremental manner: the more interfaces they expose, the more links they provide, and the more linked dataspace services they support, the more integrated into the dataspace they become.

The diagram in Figure 36 illustrates components view of the Linked Water Dataspace. We can recognize three main concepts:

1. **Datasets** such as weather data, water sensor data, building management system data, etc. Those form the actual content of the water dataspace. They are the basics for all the insights that can be drawn from the dataspace. Datasets can join and leave the dataspace. In fact, that can very dynamic such as in the case of dynamic sensor environments. Joining the dataspace requires some "cost" to be paid. This cost takes various forms: the registration into a catalog so the dataset becomes visible by others, the conformance to a schemata or the mapping to other schema, the exposure of data into a set of formats such as the RDF serialization JSON-LD, etc. We adopt in the Linked Water Dataspace a pattern in which the publisher of the data is mainly responsible for paying this cost. That is a very pragmatic feature as it allows the dataspace to grow and enhance gradually. It is a quite scalable concept, and is followed in large scale environments such as the Web.
2. **Adapters** (or Interfaces) are the technical facades of the datasets that other members of the dataspace can talk to. Such facade are a way to quantify the degree of involvement of the dataset in the dataspace. For example, a dataset that provides a JSON-LD interface allows

structured queries to be executed, and thus it is superior and more integrated into the dataspace than a document that is only exposed by keyword search.

- 3. Linked Dataspace Services** which form the backbone of the dataspace. While data is crucial for the space, support services are the platform that allows datasets to be visible, query-able, integratable, searchable, monitorable, and curatable. For instance, for a sensor to become visible to the dataspace, it must register itself in a catalog along with some information on how to get its data, how often and precise it is, etc. Such a service is provided by the CATALOG service in Figure 36.

Besides those three concepts, there is also the concept of a **Relationship** between two datasets, or between a dataset and a service. Relationships are omitted from Figure 36. for clarity, but an example would be a "replica" relationship between Druid and Data Cube for some datasets as will be discussed later on in this Section.

Applications surrounds the dataspace and make use of its support services to interact with the datasets. For instance, a realtime dashboard checks out the realtime sources from the CATALOG service, and then consume data from the EVENT service. A data analytics app discover data from the CATALOG service and runs analytics algorithm over dimensional data queried through the QUERY service.

The Linked Water Dataspace emphasizes six main services, two of them are core and the focus of this document, while the others are support services and the focus of future Deliverable D3.2. Services align with the requirements in Section 2 as shown in Table 3

Table 3: Linked Water Dataspace Support Services and Requirements

	Dataspace Service	Requirements								Enabling Technologies	
		Standardisation	Consuming Open Data	Publishing Linked Data	Data Linking	Real-time data / events	Real-time Analytics	Data integration	Heterogeneity of Sensor Data Events		Enrichment of Sensor Data Events
Core	CATALOG	+	+	+	+			+			CKAN
	QUERY	+		+				+	+		DRUID, SPARQL
Support	EVENT		+			+		+		+	COLLIDER, SPARK Streaming
	MONITORING		+			+					Event Processing Engine
	HUMAN				+			+			IMIRT Task Manager
	SEARCH				+						Entity and keyword search engines

The CATALOG and QUERY services are core to the Linked Water Dataspace as are usually emphasized in dataspace literature [59], [60]. We developed the concepts and implementations of the Linked Water Dataspace based on these two services as described in this document. Further development and addressing on the remaining services is the subject of the future Deliverable D3.2 as discussed in Section 5.5.

5.2 The Dataspace CATALOG Service

The catalog is the central registry of the dataspace. Within the catalog all datasets and data sources are declared along with several meta-data about them. This includes (i) the list of entities managed by the system such as sensors or locations and (ii) open data sources that are relevant to water management such as weather observation stations or forecast services.

More specifically and as illustrated in Figure 37, the Waternomics catalog service gives access to data sets from various sources:

- Historical Sensor data from NEB (Galway pilot) BMS system originally store in SQL database.
- Real-time VTEC sensors installed in NEB
- Aggregated data from OpenCube
- Open Data such as the Weather Data
- Real-time data from other pilot sites that are sent to the dataspace through a RESTfull API ¹ to the Kafka middleware.

The Linked Water Dataspace is built upon the CKAN dataset portal. In essence CKAN is made for creating a data portal that is open and available online. In the Linked Water Dataspace CKAN has been extended with the ability to manage water entities such as sensors, locations and users. Registering at the catalog is the first sign of a dataset that joins the dataspace as described in Section 5.4.

Figure 38 shows a screen capture of the Linked Water Dataspace. A search request with the term “Galway” reveals 4 data sets. We describe here only the first 3 data sets as they appear on Figure 38. The first one concerns the NEB (Galway pilot) water sensors. The data format for this dataset is SQL, indeed, this describes the access to the NEB SQL database for collecting water sensors data. The second data set relates to the Galway city current weather observations taken from openweathermap.org. Access to the data, as described in the portal, is done through http access and the returned data is in JSON format. The third data set is the weather forecast for Galway from met.no. It describes and provides the API link to the weather forecast data to be collected by relevant open data services.

¹ Example can be seen here: <http://linkeddataspace.waternomics.eu:8003/message?topic=vtec.eindhoven.json>

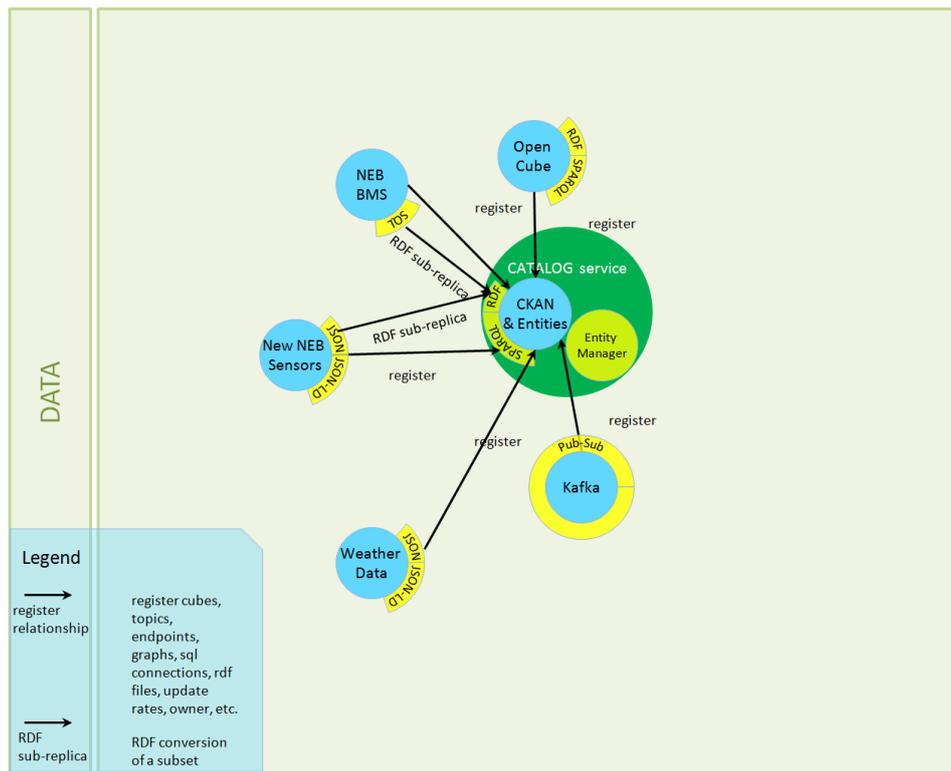


Figure 37: Linked Water Dataspace- The CATALOG Service View

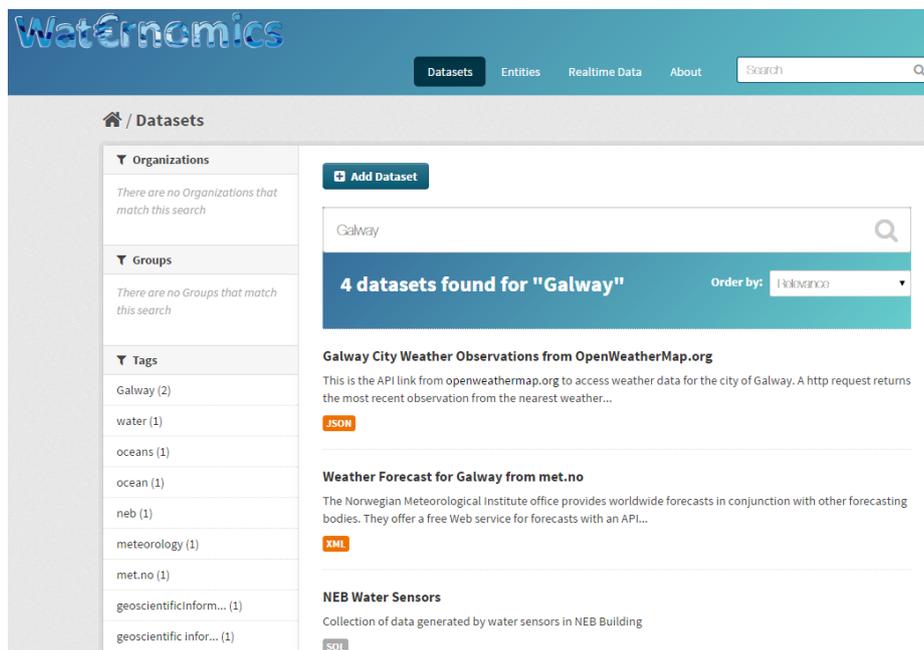


Figure 38: Waternomics Dataspace – Data sets related to Galway

Details about these data sets are also available. For example, Figure 39 shows the NEB historical logs data set description. This description reveals the URL to access this data with additional information such as the data entered to the dataspace, its status, etc.

An important feature for the catalog within Waternomics Linked Dataspace is the ability to manage entities. Managing information about the critical entities in an information system is one of the primary requirements. This is due to the fact that all other components rely on this information for their functionality. Understandably, the real-time Linked Water Dataspace

(LWD) is no exception to this requirement. The entity management process is concerned with the maintenance of information about entities critical to the Water data management and analysis. The expected outcome of this process is a database that serves as the canonical source of metadata for sensing data. In the case of Linked Water Dataspace, the primary set of entities includes the sensors and their physical locations.

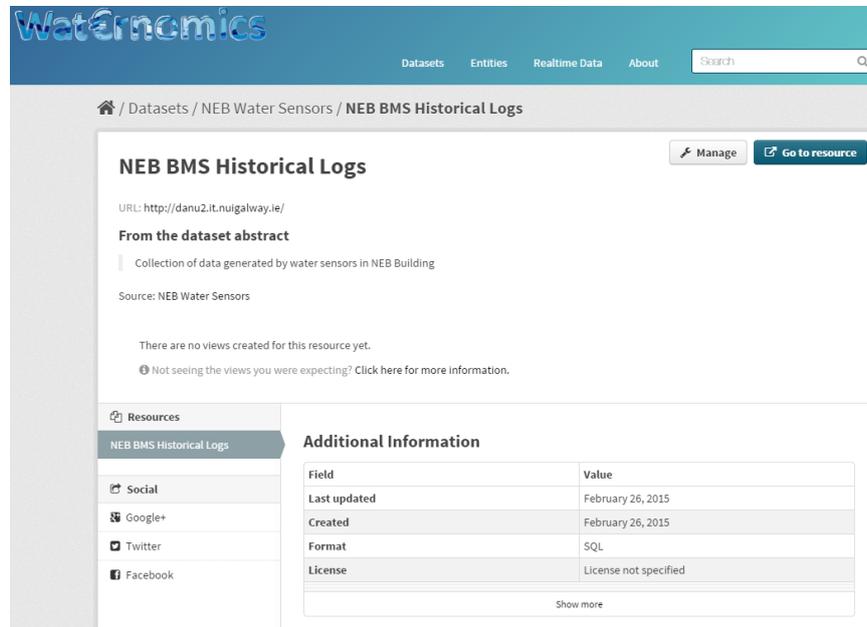


Figure 39: NEB Historical Logs in Waternomics dataspace

Beside these entities, the dataspace applications might also require information about the people, groups, buildings, and outlets. In short, all of the information that can help in understanding the water consumption, through association with real-world objects, is included in the entity management process. However, the level of quality control might be differentiated depending on the criticality of entity for dataspace applications. This highlights the need for appropriate entity management process both in terms of software tools and operational guidelines. In the rest of this section, we start by listing the entities that are required to be managed in the LWD. Then, we summarize the expected sources of entity information in the LWD. The entity management process workflow id the focus is partially addressed with Section 5.4 on joining the dataspace, but will be further investigated through crowd sourced entity management, represented by the HUMAN dataspace service, and discussed in the future Deliverable D3.2.

5.2.1 Managed Entities

In the Linked Water Dataspace the most important entities are concerned with the sensors and their locations. Additionally, the information about water outlets can also be useful for applications. In the following we describe the entities and their minimal set of attributes required in the LWD.

- Sensors:** The sensors that measure the flow of water generate the streams of data at different intervals. This data is used to calculate the water consumption levels of the area covered by a sensor. Generally, different types and forms of sensors are installed for metering water consumptions. Therefore, it is necessary to exactly describe each sensor, its capabilities, and its coverage. The sensor description may include the identifiers and labels. The sensor capabilities can include information like the units of measurements and rate of measurement. In Table 4 we describe some example attributes of a sensor entity.

The attributes are stated in the RDF using the Dublin Core Metadata Initiative Terms (dct) vocabulary.

Table 4: Required attributes for the entity Sensor

RDF property	Example Value
dct:type	water:Sensor
dct:identifier	12
dct:title	Sensor 3
dct:description	Sensor 3 on Second Floor
dct:coverage	F2

- Outlets:** Besides sensors, information about the actual physical water outlets is also important for analysis and decision making. It is possible that a single sensor might be installed for a set of outlets. In such cases, a cumulative assessment for water consumption is needed. Table 5 shows the minimum set of attributes required for each entity that represents an outlet.

Table 5: Required attributes for the entity Outlet

RDF property	Example Value
dct:type	water:Outlet
dct:identifier	CWTF2
dct:title	CW Tap - Floor 2
dct:description	Cold water tap on Floor 2

- Sites:** Information about sensors and outlets is not useful without the information about their associated spatial locations. For instance, it is difficult to perform meaningful analysis when only the water consumption of a pipe or a tap is known. Water flow of a pipe along with the locations serviced by the pipe constitutes more actionable information. Therefore, we consider sites as the third critical set of entities in the LWD. For simplicity, we assume that each sensor covers one or more sites and each outlet is installed at single site. Table 6 lists the minimum set of attributes required for each site entity.

Table 6: Required attributes for the entity Site

RDF property	Example Value
dct:type	water:Site
dct:identifier	F2
dct:title	Floor 2
dct:description	The Second Floor in NEB

5.2.2 Information Sources

The information about entities is generally spread across various systems in an organization. The LWD is no exception to this observation, but majority of data about primary entities can be found with the facilities management department or a similar unit. Building management systems (BMS) are a prime example of such information source. Modern BMS include variety of sensors and actuators to control the behaviour of a building depending on usage and environment. If available, the BMS can be considered the main sources of information about

sensors, outlets, and sites. However, this assumption might not be correct in various cases. It is not common to have BMS installed for large and old buildings. Similarly, airports and cities also lack the infrastructure required to install a BMS. In such cases, it is common to have individual databases for sensors, outlets, and sites. Each of those databases might have different data formats and management processes. It is essential to have such information integrated and stored in a common format with same semantics.

The Linked Water Dataspace consisted of information related to the water consumption as well as the information that supports the decision making process. In this regard, information from other sources is brought within the dataspace. Such sources generally exist outside of the organization and require source specific data collection processes. For instance, it is now common for the weather websites and meteorological to provide interfaces for exporting their data. The interface can be in the form of RESTful API or comma separated text files. Essentially, these sources provide information about entities which might be critical to dataspace but provide useful contextual information. For example, an open day at the university might indicate the higher than usual water consumption. A BMS does not have such events information in its database. In short, applications might require entity data from external sources as well.

So far we have discussed the entity information that is stored in databases. However, the information contained in databases might become inaccurate with the passage of time. Most importantly, the databases might not contain the complete information about entities. To counteract such issues, it has been proposed to include the users of the system in the entity management process. However, this is not a trivial task which might need careful consideration. Nonetheless, inclusion of the user community in the information curation process not only helps in maintaining accurate data but might also help in building user trust in, and ownership of the system. The community of users can help describe additional properties of entities such as the working state of water outlets. An appropriate process is required to source data from a community of users which is the focus of the HUMAN dataspace service discussed in the future Deliverable D3.2.

5.3 The Dataspace QUERY Service and the Realtime Aspect

The architecture for the Linked Water Dataspace shown in Figure 36 is designed to meet the requirements for integration of multiple sources for water management, with various levels of data updates, coming from legacy data or from sensors. The QUERY service as defined here covers various aspects and processes of the dataspace that lead to the data being structuredly queried by the applications. The QUERY service also addresses low latency and fault-tolerant data analysis. The proposed architecture to realize The QUERY service follows the Lambda Architecture recommendations [45].

The Lambda architecture [45] is a concept which is getting acceptance by Big Data researchers and practitioners. Herein, the Lambda architecture realises the need for integrating water data within a data warehouse that is processed by batch processes and views that are pre-computed for fast access by applications. The Linked Data principles serve as a mediator to improve the integration within the batch layer for relatively not fresh data.

Besides, the architecture realises the need for a real-time aspect of the water data. Such an aspect could be crucial to support decisions relevant to fast detection of situations of interest such as leakage, and react accordingly. That is achieved by the speed layer that effectively works over streams of linked water data. Streams are not actually stored but rather processed in flow to guarantee a low latency view of the data that can complement the older views achieved by the batch layer.

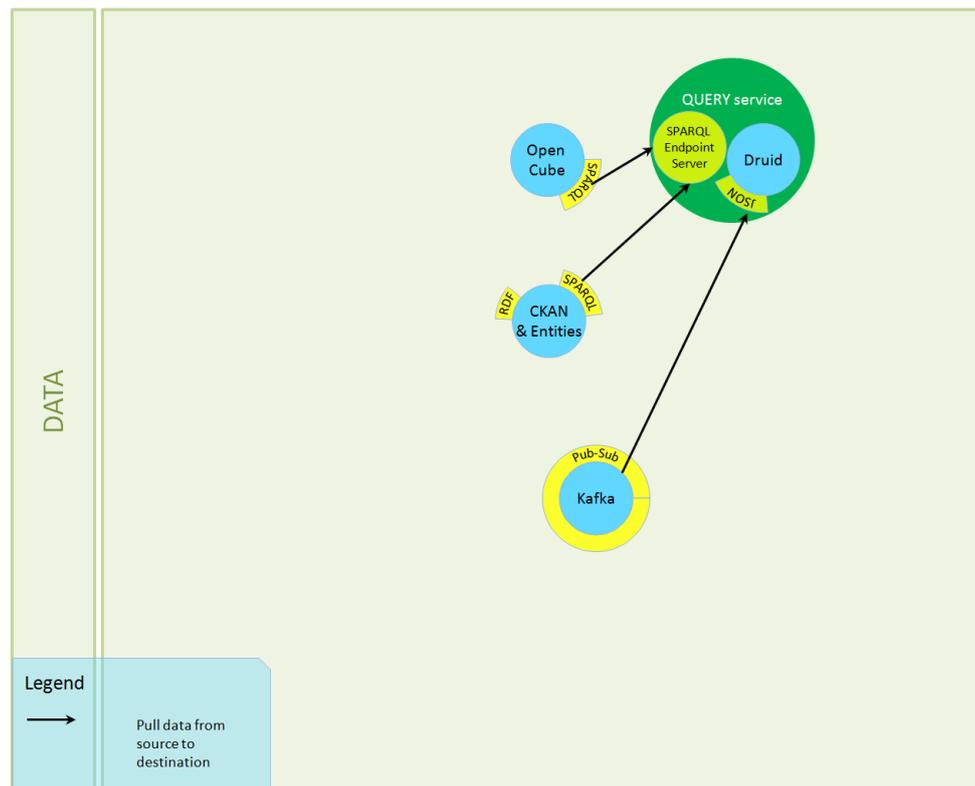


Figure 40: Linked Water Dataspace Lambda Serving Layer-- The QUERY Service View

To provide applications with a single interface for data access, a serving layer is provided. This forms the query interface for the applications and is the visible facet of the dataspace QUERY service as shown in Figure 40. This layer splits queries to the batch and speed layers to combine pre-computed results from the batch layer, with near-time fresh and maybe approximate results from the speed layer.

The query results are combined and transparently returned to the user. The following components are the defining keys of the Water Lambda Architecture as will be discussed further in Section 6:

- **Water Datasets:** those include the building management systems, legacy data such as water consumption logs or bills, as well as sensors installed on the pipes for instance.
- **Adapters:** those serve as the first step to normalise the water data into the Linked Data realm. They respect the semantic model and convert data items such as sensor events into their corresponding canonical linked data form. Details about the first version of the vocabulary used in this project are discussed in Appendix A: Minimal Vocabulary.
- **Management layer:** it is responsible for providing the functionalities and services essential for managing metadata such as ontologies as well as entity data such as people, places, etc. This layer is handled within the CATALOG service as discussed in Section 5.2.
- **Batch layer:** it provides batch-based processing of the Linked Dataspace for accurate, but with some delay, data views, such as aggregation-based water data views, and basic analytics. This layer is exemplified in the processing of water consumption data from the building management system as discussed in Section 6.1.

- **Speed layer:** it provides real-time processing for water data with low latency processing such as approximate event matching, data enrichment and complex event processing. Details about real-time sensor data processing are provided in Section 6.2.
- **Serving layer:** it provides a transparent query of data from batch and speed layers, along with mid-level services such as activity detection and prediction services. This serving layer is realized via Druid and described in high level in Section 6, while it will be detailed as part of future deliverables D3.2 and D3.3.
- **Applications:** applications interact with the architecture via the serving layer and gain access to multiple water data views so their processing can take place starting from there, either of simply presenting data in user interfaces, or providing further processing capabilities and control mechanisms. Applications also will be covered by a future deliverable D3.3.

An important aspect of the QUERY dataspace service is making data available in the statistical form through Linked Open Data principles. This is similar to availing a data warehouse query end to applications. This is done via the OpenCube platform which is based on the Open Data principles. Open Knowledge Foundation¹ defines Open Data as: “*data that can be freely used, reused and redistributed by anyone - subject only, at most, to the requirement to attribute and sharealike*”.

It means that particular data should be available free of charge to everyone for the personal as well as the commercial usage without any restrictions. This idea is not new and is similar to the “open” movements such as Open Source or Open Hardware.

The growth of the World Wide Web, Internet and easier access to the web allowed the Open Data movement to gaining the momentum. Moreover, some of the governments are willing to make their data available to the public in open formats. This ensures the government transparency and increases the trust. Furthermore it has an impact on the development of services and applications addressing public demands. The best-known examples of government open initiatives are data.gov and data.gov.uk.

The Waternomics platform should be able to make use of relevant Open Data assets for proper analytics. Possible scenarios for consuming Open Data include the prediction of water consumption using open weather data. Data consuming requires a proper selection and evaluation of data source in order to select the most suitable one for a proper decision support. The more “linked” datasets are available, the more accurate the predictions and analysis can be.

After analysing the existing and well-established Open Source projects that supports the operations on the Linked Data, in particular the Linked Open Statistical Data in Section 4.2, we notice that the described platforms may be used as an element of Waternomics Batch Layer, which provides batch-based processing of the Linked Dataspace for accurate data views, such as aggregation-based water data views, and basic analytics, with some delay.

The RDF store should support batch ingest capability with speed. The number of entities is expected to be large, so the Batch Layer should be fault free and relatively simple.

We propose to use OpenCube toolkit as it proposes a complete set of tools that can be reused and adapted for the Waternomics project.

¹ <http://okfn.org/>

5.3.1 Data Cube Vocabulary

Statistical data is modelled as data cubes, which describes the multi-dimensional data. Since 2014 W3C Data Cube Vocabulary¹ is a W3C standard for modelling multi-dimensional data as it is the most popular and stable vocabulary (see vocabulary in Figure 41). Data cubes are characterised by the dimensions and the measures. As shown in Figure 41 Data Cube Vocabulary allows additional attributes. The cube dimensions define what each observation is about, while the measure defines what is being measured.

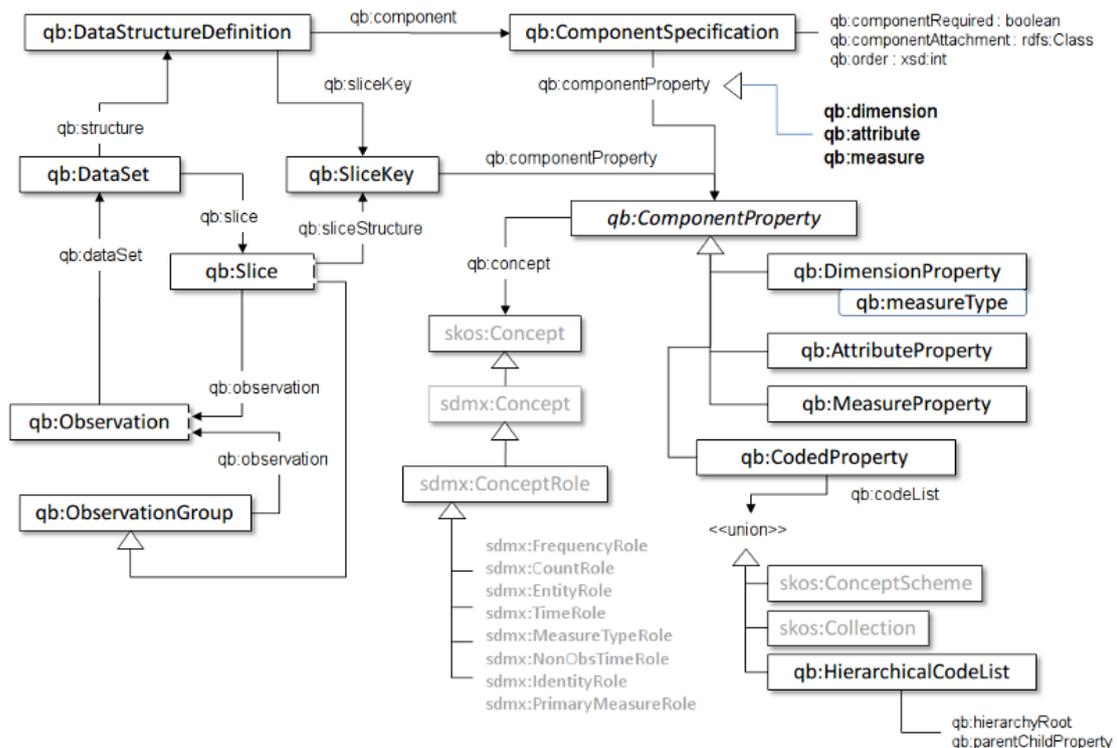


Figure 41: Outline of the Data Cube vocabulary²

5.3.2 Data quality and Data Linking

Data linking is the fundamental idea of the LOD as described in Section 4. The goal of the Waternomics project is to allow linking of the sensors data, data management systems and water meters using the benefits and potential of Linked Data.

Water data published to the dataspace becomes more valuable when it is linked to other data sets. This linking is very useful for ensuring an optimal data management and integration. It helps enhancing its re-use and allows discovering new knowledge from water data. It is important to determine what data sets are relevant to be linked with water data.

Developed linked dataspace will be able to capture and store data from various sources. To ensure the high quality of the data, the following functionalities need to be supported:

¹ <http://www.w3.org/TR/vocab-data-cube/>

² <http://www.w3.org/TR/vocab-data-cube/>

- **Data refining:** Data refining is a process of data cleaning and normalisation to reduce information duplication. It is very important process, as the distortions at the input data affect the results of the analysis. Data refining removes the data variability and redundancy. As a result it develops an integrated data resource.
- **Data enrichment:** This functionality aims at enriching the unstructured content with additional content from other data sources. In Waternomics implementation we may link the sensors data with i.e. geographical, facts and entities datasets (places, organisations, things).
- **Data interlinking:** It is a process of creating links with relevant web entities. This activity reveals relations with the various topics published on the Semantic Web.

5.3.3 Data exploration

Data exploration is a feature of the platform that allows the navigation through the metadata of the collection of datasets at the knowledge base. The most important features include:

- **Data Catalog** – this includes a user interface that allows browsing through the collection of datasets. It should include browsing by categories and all type of sorting and filtering.
- **Searching** – this includes full-text search over the data, as well as the metadata. Also it may include vocabulary search.
- **Graph-based navigation** – is a way to navigate through the related datasets and triples. It may be implemented i.e. by the widgets displaying related entities.
- **Data Discovery** – it a feature that allows finding the relationships links between entities in related RDF data.

Data exploration is very important part of the system, as it allows finding new insights and relations between the data.

5.3.4 OpenCube for Water Data

At the moment OpenCube Toolkit seems to be the best candidate for the Waternomics batch layer needs. The platform will be used for exposing the sensors and water data as cubes of Open Data. Currently the tool supports both: the publishing and the reusing phase of the Linked Data lifecycle.

5.3.5 Data Publishing

Data publishing is implemented by the Data Provider¹ concept. The platform gathers the data according to the provider configuration and stores it at the data store. The import / conversion process can be scheduled. The platform allows data importing from tabular data (such as CSV/TSV files), relational databases and XML files. RDF data can be uploaded directly to the system or imported from selected URL or SPARQL endpoint.

For the Waternomics needs the TARQL component seems to be the most useful. It is integrated with the OpenCube Toolkit as a data provider (see Figure 42). It enables data cubes construction

¹ <http://sdk.fluidops.net/resource/Help:ProviderSDK>

from CSV/TSV files. During the OpenCube project the TARQL Open-Source technology was redesigned (API) and the streaming evaluation mode was included.

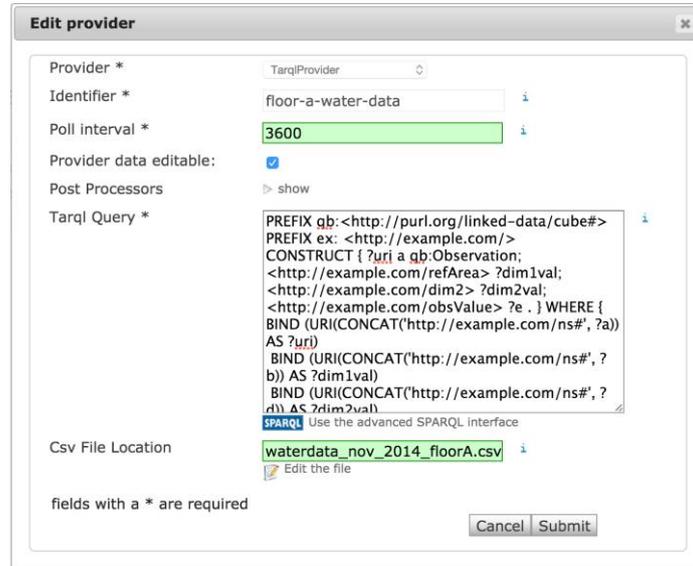


Figure 42: TARQL Data Provider configuration screen

5.3.6 Data Reuse

The Waternomics system should support integrated analytics of both real-time and historical data for effective decision support. For that purpose OpenCube Toolkit provides various widgets:

- R statistical analysis module: Based on the input data, parameters and R script it generates a chart in R. The result can be displayed as the image on a wiki page. The widget provides full support of the R language.
- OpenCube Browser: It is a table-based visualization of RDF data cubes and exploration of an RDF data cube by displaying a two-dimensional slice of the cube as a table. It allows using the data stored in the platform as well as the data from remote SPARQL end-points. The user defines the 2 dimensions that are included in the table of the browser.
- OpenCube Aggregation component: The role of this component is to precompute aggregations of the data cubes in order to enable OLAP operations. As a result it creates a set of Data Cubes out of an existing cube by summarizing observations across one or more dimensions. The results can be used by the other components i.e. OpenCube Browser.
- Interactive chart visualization widgets: This component allows visualization of the RDF data cube slices, by using chart-based functionality as shown in Figure 43.

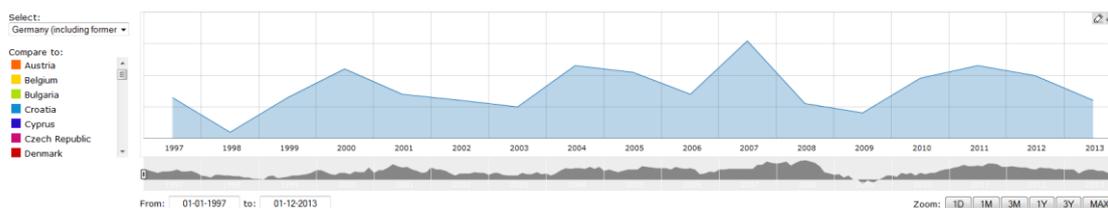


Figure 43: OpenCube Chart Visualization widget

5.4 Joining the Dataspace and Adapters

The Linked Water Dataspace is composed of multiple data sources. The types of data sources can include but is not limited to sensors, databases, text files, and Excel sheets. To enable a data source to be part of the Linked Water Dataspace, it must be discoverable and should conform to the 5 stars scheme of Linked data. The critical entities in the data source should be exposed as Linked data. We propose a seven step approach for managing entities of a data source according to the principles of Linked Data. The approach allows the conversion and publishing that managed entities in standard Web formats for Linked data such as RDF and JSON-LD. It further facilitates the open availability of entity data in support of various applications. Figure 44 provides an overview of the joining process for data sources.

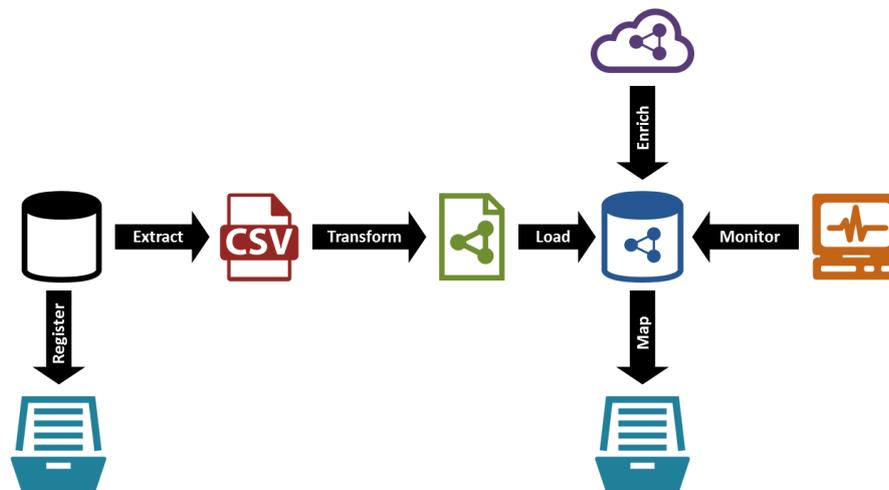


Figure 44: RETLEMM process for exposing entities as Linked data

- **Register:** A new data source joining the dataspace would require it to be registered in the dataspace catalog. The registration means that the catalog contains an entry describing the data source at minimum in terms of its type and its format. Further information about the data source can include the physical address of files or a query endpoint. After this information a dataset or data source is considered part of the dataspace, since it can be accessed and used by application.
- **Extract:** The second step of the management process is to extract the entity information from the data source. For the sake of simplicity, it is assumed that such information will be extract in the form of CSV¹ files. Besides the data extraction from databases, this step can also include data collection from community of users. Essentially, the purpose of this step is to source relevant information about entities from either digital or human sources.
- **Transform:** Given the CSV representation of entities, the next step is to convert the data in appropriate format for publishing. In this regards, we propose to adapt a simple semi-automated process for transforming the CSV files to RDF files using appropriate tools such as Microsoft Excel and Google Refine. This process is supported by the schema mapping document that define the correspondence between the attributes in sources with the RDF properties in the target Linked Data representation of entities. It is recommended to map each source attribute with a property in WATER or DCMI vocabularies that are defined later in this report.

¹Comma Separated Values

- **Load:** Once the data has been converted and represented in RDF format, the next step is to store it in an appropriate data store. For this purpose, any general purpose triple store may serve the purpose. However, it is necessary for the RDF store to have necessary publishing, querying, and search functionality to support applications. Additionally, the RDF store should support batch ingest capability. Since, the number of entities can be large the batch ingest should be fault free and simplistic. We recommend following the batch layer approach of the Lambda¹ Architecture for this purpose.
- **Enrich:** Generally above steps are sufficient to support analytical and decision support applications. Nevertheless, it is desirable to enhance the metadata with additional information such as links to external datasets and additional attributes extracted from non-critical data sources. Therefore, the purpose of this optional step is to add additional RDF triples to the store so that the overall vision of the Linked dataspace supported. Essentially, this step should be driven by the needs of individual applications to support their functionality.
- **Map:** Similar to enrich step the entities and schema of a data source may be mapped with other data entities and data sources in the catalog. This facilitates integration and de-duplications of classes and entities. Additionally, it allows automated process of data collected from multiple dataset using advanced reasoning and schema agnostic query tools.
- **Monitor:** As discussed earlier, the entity metadata is considered to be immutable and append only. However, it is not unusual for the organizations to change or update the definition and attribute of their core entities. We propose to hand this challenge through an appropriate monitoring process. As this point in time, we keep the monitor process to simple scheduled checks for changes in attributes types and values. We expect this to change as the project matures and more insights are brought in terms of the data management process.

At this end, the process applications developers should be able to find and understand the data source. Generally when a data source joins the above process can be performed manually by the data source owners with the help of dataspace administrators. However, the ETL step can be automated to speed-up the process. In the following, we discuss the two alternatives for automation:

- **Adapters:** Adapters can be considered non-materialized view of a data source. They encode the ETL process in the form of mapping queries between source data format and the target data format. Essential, the data resides in the source the ETL is performed in the fly every time a request is posted for entities.
- **Scheduled Jobs:** The ETL is process is performed after pre-determined intervals to keep the entities in source and the triple store synced. This also allows reflecting any changes in the data source in terms of addition, deletion, or update of entity metadata.

5.5 Conclusion and Plan

In this section, we have provided an overview of the Linked Water Dataspace. The proposed architecture is first glance at how the dataspace will be realised. We propose to follow the Lambda architecture and extend it to facilitate the Linked Data approach. The dataspace is used for ingesting, managing, and publishing water data that includes both sensor data and entity data.

¹<http://lambda-architecture.net/>

First, a simple entity management process is proposed that exploits non-technical experts. Second, a complex event processing approach is proposed for real-time sensor data. This approach enables semantic processing of in-flow data enrichment and approximate matching. Third, we propose to employ OpenCube toolkit to publish aggregated sensor data as open data. In short, this section provided a glimpse into the implementation of Linked Water Dataspace with respect to the requirements introduced in Section 2 as shown in Table 7.

Table 7: Summary of Requirements and Approaches

Requirement	Approach
Standardisation	Standard compliant ontologies such as SSN will be used for describing sensors and their data.
Consuming Open Data	One of the support services that will be included in the dataspace consists of crawling weather data an integrating it in prediction services.
Publishing Linked Data	The produced data by every support service will be published in RDF using linked data principles.
Data Linking	Relevant open data sources will be integrated in the dataspace through explicit links between entities.
Real-time data / events	The dataspace is design with respect to the lambda architecture that covers real-time event processing. We propose a complex event processing engine for managing live sensor events.
Real-time Analytics	Real-time analytics of data is considered in the platform as part of the speed layer of the lambda architecture.
Data integration	Lambda architecture in essence was proposed to deal with both historical and real-time data. This data integration is insured by relevant support services.
Heterogeneity of Sensor Data Events	The platform is designed to handle a variety of sensor types. Consequently for each sensor type, a dedicated collector is designed for collecting data and converts it into RDF for further processing.
Enrichment of Sensor Data Events with Open Data	Sensor readings are further enriched by dedicated support services using open data.

While the basis and infrastructure have been developed for the Linked Water Dataspace as discussed in this document, this work is in progress with plans put to complement the Linked Water Dataspace in D3.1.2 in five main directions:

- Core Dataspace Services:
 - a. The CATALOG service: we plan to investigate the suitability of the CKAN framework for playing the role of a catalog for the Linked Water Dataspace and whether an extension shall be developed. One aspect is whether it can provide a granular catalog for entities not just for datasets. Another aspect relates to the ability of the catalog to host master data, stick to a replica-based paradigm, or host links to actual data and support a re-direction of queries and requests to the datasets. The last aspect would be the investigation of the role of the catalog as a guide to real-time

datasets which volatile after sometime and what would be the best way to catalog such data source.

- b. The QUERY service: applications may query dimensional data from Druid or from the OpenCube as RDF. We plan to investigate other types of data which are not purely dimensional but rather could consist of linkage between entities, real-time, and legacy data at lower levels of granularity. Other investigations would include the suitability of the Druid deep storage for Linked Water Data, with possibilities to adopt other deep storage mechanisms.
- Dataspace Support Services:
 - a. The EVENT service: the event processing service is planned to be added to the dataspace to meet the requirements of Heterogeneity of Sensor Data Events, and Enrichment of Sensor Data Events with Open Data based on models such as approximation and dynamic enrichment [62]–[65].
 - b. The MONITORING service: the updates of the dataspace and propagation of such updates shall be investigated within the support services realm. The suitability of the publish/subscribe middleware will be investigated with focus on integration with the CATALOG service.
 - c. The HUMAN service: inclusion of the community of users and possible crowds into the entity management process is one possible approach to reducing costs and time of entity management. This approach requires outsourcing of specific entity management operations to non-technical people. In case of crowds, these operations, such as providing the location of a sensor, will be performed by people who do not have the database management expertise but do have the basic skills to perform repeated tasks.
 - d. The SEARCH service: to ease the accessibility to data in the dataspace, users and developers may perform keyword-based search over the datasets. We plan to investigate this requirement within the Linked Water Dataspace and develop a suitable and effective method to shorten the time needed by users and developers to reach the data in relevance to their needs.
 - Ontology and Data Models: ontology developed in Appendix A is an on-going work which requires refinement. That is planned through versioning and as a result of validation that can become apparent throughout the integration process between the various Linked Water Dataspace components and integration with other pilots.
 - Pay-as-you-go Integration of Datasets: we plan to enhance the involvement of datasets and sensors into the dataspace by providing new interfaces (adapters), providing links to other datasets, conforming to a common data model, and mapping between entities or schema.
 - Evaluation and Validation: The evaluation and validation is an on-going process which targets testing various qualitative and quantitative aspects of the dataspace. We plan to test the suitability of the data model and the catalog to meet various types of sensors and datasets in the water space. Also the effectiveness and efficiency of various dataspace services including query time, event processing latency, precision, recall, throughput, and other factors of the support services will be investigated.

6 Concrete Cases

The architecture has been implemented mainly through an Apache stack of technologies as shown in Figure 45.

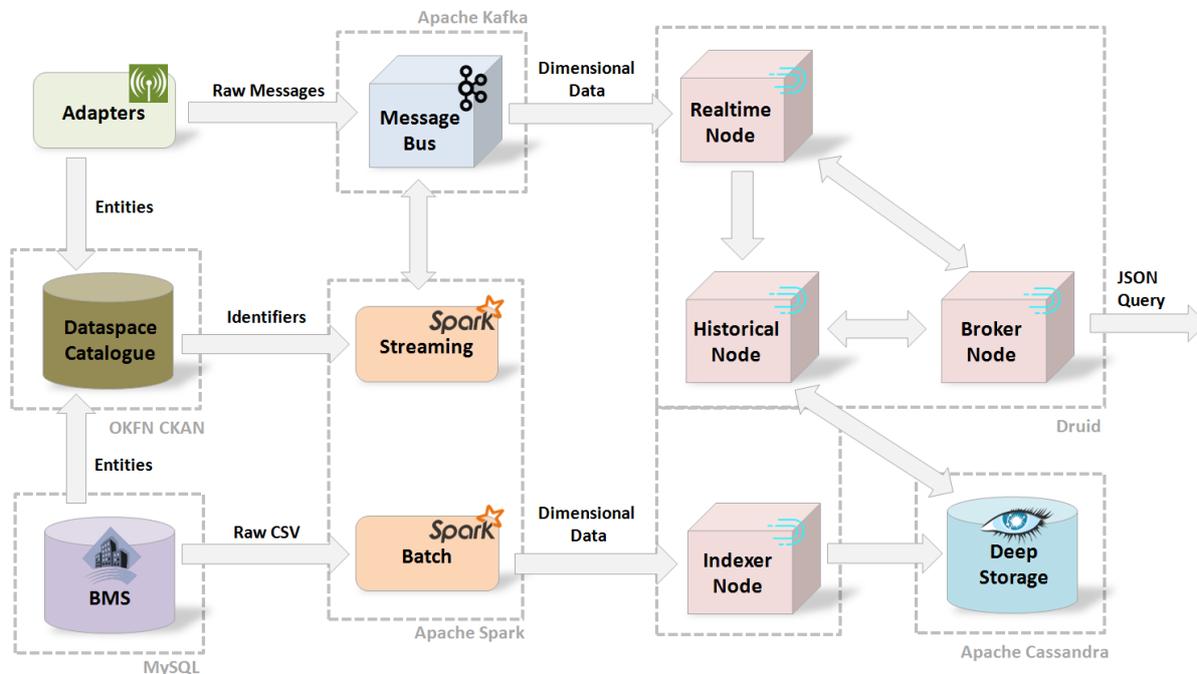


Figure 45: Linked Water Dataspace Lambda Architecture Realization

Figure 45 shows that entities from the BMS are exported into the CKAN dataspace catalog as well as entities from other sources such as newly installed sensors. Batch data is fed then into map/reduce jobs in the Spark SQL node, while real-time data from sensors is fed into the Kafka message bus, widely used for Big Data distribution and then into map/reduce jobs in the Spark Streaming node. The result from the batch nodes fed into a Druid indexer node as dimensional data, while the streaming data goes into Kafka and then into a Druid real-time node. The Druid nodes could use a Cassandra deep storage data store. Both batch data and real-time data are exposed transparently via a Druid broker node which can be queried by applications in JSON format. We present in the following two sections two concrete cases that have been implemented within the Linked Water Dataspace in NUIG being: the building management system, and a water sensor joining the dataspace.

6.1 The Building Management System Joins the Dataspace

The New Engineering Building (NEB) at NUIG has a commercial grade Building Management System (BMS) that collects and stores data from existing water sensors. The NEB BMS contains information about events and entities. Entities include such as sensors, rooms, water outlets, etc. Events include metering data generated by sensors which records information such as water flow. The NEB BMS joins the Linked Water Dataspace by following the process described in Section 5.4, which enable the visibility of entities in dataspace. The events data that is collected over many months is included in the dataspace through batch processing, as shown in Figure 46.

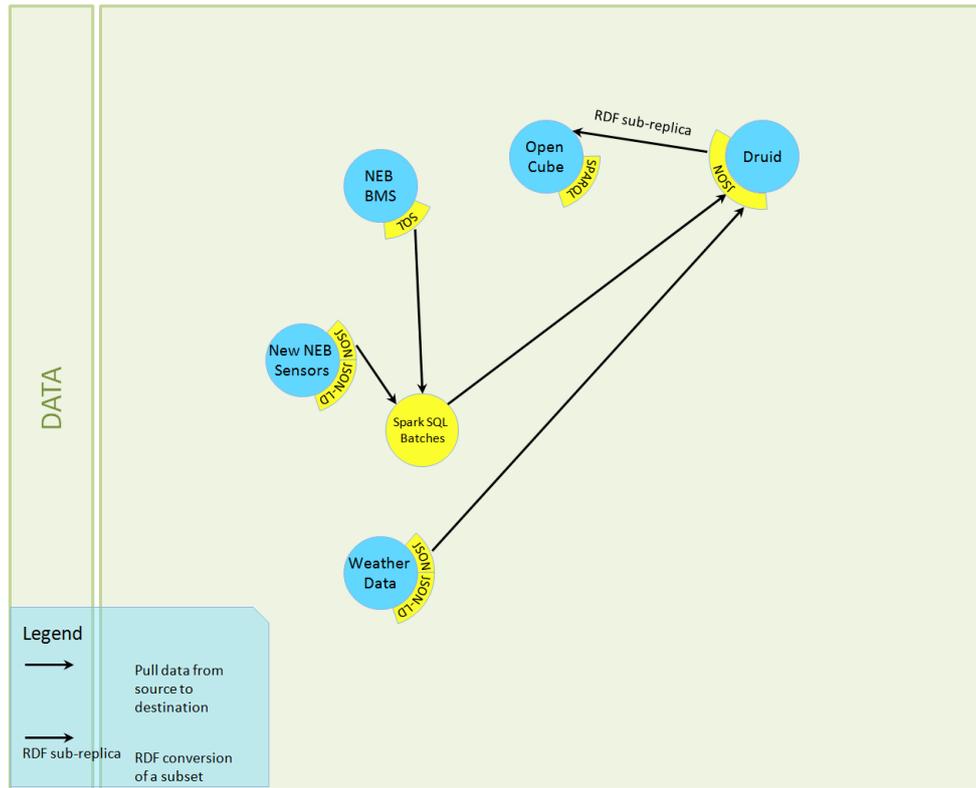


Figure 46: An example realization of Lambda Batch Layer in Linked Water Dataspace

The batch processing aspect of the dataspace is realized via the following elements:

- A set of SQL scripts are used to extract the historical events data from the database of NEB BMS.
- A Spark Core map/reduce jobs is used to transform and aggregate low levels events to suitable granularity.
- A Druid indexing node is used to move the dimensional aggregated historical data into the Druid deep storage and make it available for further querying.
- The aggregated dimensional data is further replicated in OpenCube expose it as Linked data.

The above process is performed once for ingestion of large number of events collected over many months. Afterwards set of Spark scheduled jobs will be deployed to ingest daily events from the NEB BMS.

6.2 A Water Sensor Joins the Dataspace

The real-time aspect of the dataspace is realized via the following elements:

- A RESTful API adapter for the sensor.
- The Kafka middleware which forms the backbone of events distribution and reliability guarantee for production/consumption staging.
- The Spark Streaming map/reduce jobs which do the first shot of aggregation and processing.

- The Druid real-time node, which moves the dimensional aggregated stream data from the Kafka bus into the Druid deep storage and make it available for further querying.

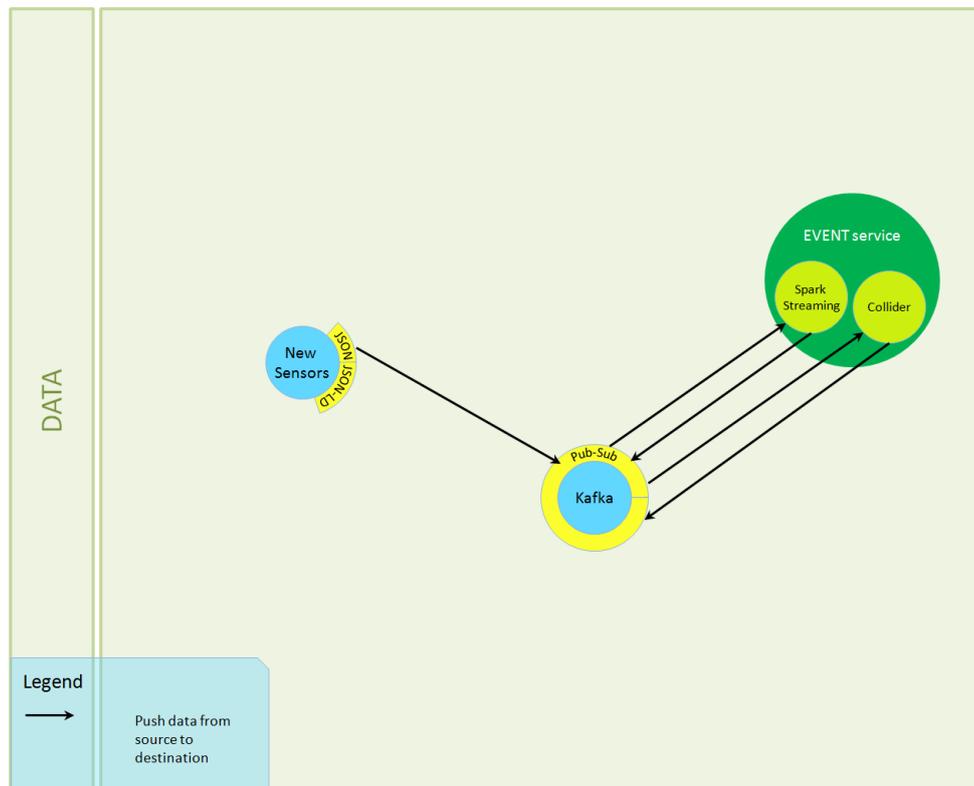


Figure 47: Linked Water Dataspace Lambda Speed Layer

The new sensor installed by VTEC in the NUIG engineering building is capable of producing JSON data that describes the readings associated with the sensor at a specified timestamp. The following Listing show an example JSON data from the sensor.

Listing 1: Sensor JSON Data

```

{
  "href": "#",
  "SensorID": "IRE01",
  "description": "This is NUI Galway Engineering Building",
  "date": "15-02-24",
  "time": "11:16:07",
  "data": [
    {
      "type": "fVel",
      "metric": "m/s",
      "value": "1409.03"
    },
    {
      "type": "today",
      "metric": "m3",
      "value": "5.99064"
    },
    {
      "type": "net",
      "metric": "m/s",
      "value": "257"
    },
    {
      "type": "flow",
      "metric": "m3/h",
      "value": "0.775783"
    },
    {

```

```

    "type": "vel",
    "metric": "m/s",
    "value": "0.17148"
  }
}

```

The JSON data is sent to the Linked Water Dataspace via a RESTful API as HTTP POST to <http://linkeddataspace.waternomics.eu/sendsensordata>. The RESTful API implemented with the dropwizard-kafka-http client then forwards the JSON messages into a Kafka topic. Other processes can consume from the Kafka topic, including an adapter process that performs a JSON to JSON-LD RDFization of data according to the ontology.

Another important process for aggregation is implemented with Spark streaming. Spark streaming jobs are map/reduce jobs following a distributed processing model. For example, because the event does not contain the absolute water consumption between two time points, the Spark streaming job is responsible for producing a stream of events with such absolute values by processing the raw "today" value in the JSON data and calculate the successive differences.

Another Spark streaming job aggregates the data by minutes as several readings can arrive within a minute. The result is a dimensional event of the form shown in the following Listing:

Listing 2: Sensor Dimensional JSON Data

```

{
  "year": "2015",
  "month": "2",
  "day": "24",
  "hour": "11",
  "minute": "16",
  "sensor": "IRE01",
  "consumption": "0.00053",
  "metric": "m3"
}

```

The new JSON dimensional aggregated data is published to a new Kafka topic. A Druid realtime node is configured to persist this data into the deep storage and make it query-able in a similar way the batch data is, thus applications can query the batch and speed layer via the serving layer and make use of it either for presenting the data visually or conducting various types of analytics over the data.

7 Summary

Water management is a challenging issue with the growing demand on water resources due to the increased urbanization, climate change, economic growth, etc. In this context, Waternomics project aims to develop and introduce ICT as an enabling technology for managing water as a resource and increasing end-user conservation awareness.

A major step towards achieving Waternomics objectives consists of collecting water usage data coming from sensors together with other relevant open data sources for an effective analytics to drive decision making: e.g., planning, adjustments and predictions.

Relevant data collected from various sources need to be standardised, enriched interlinked and shared among services and applications. We call this process: Linked Water Dataspace management. This deliverable reports on the initial design efforts towards this Linked Water Dataspace with respect to the requirements defined in Section 2.

After analysing relevant technological contributions that can be used in this context (see Section 4), we propose to design a dataspace following principles of the lambda architecture that facilitates the management of both historical and real-time data. The result of our research is reported in Section 5.

We propose in this project to reuse existing open source tools that have been proven to be effective in practice. Indeed, the availability of high quality Open Source tools that support the process of publishing data as Linked Data, reduces the cost of implementation. By making the water data available as Linked Open Data new opportunities arise. Data may be combined across multiple datasets and manipulated in a more efficient way.

The primary contribution of this report is a proposed architecture for implementing the Linked Water Dataspace. The proposed architecture is aimed at managing water and other relevant open data and exposing it as Linked Data. While the basis and infrastructure have been developed for the Linked Water Dataspace as discussed in this document, this work is in progress with plans put to complement the Linked Water Dataspace in D3.1.2.

The rest of this document consists of two appendices. Appendix A represents a minimal vocabulary for describing sensors and sensors' readings. It is based on an analysis to the existing relationship model provided in Deliverable D1.3 and the mapping of them into ontological concepts. Appendix B represents a brief survey of open data for water management. It constitutes our initial efforts towards the investigation of relevant open data sources that are relevant in the context of water management.

Appendix A: Minimal Vocabulary

This appendix reports on our efforts towards designing the vocabulary that we are using for modelling sensors and their observations in the proposed dataspace.

1. Methodology

In order to develop the required minimal vocabulary for describing sensors and their readings, we proceeded in two phases of work. As shown in Figure 48, phase 1 consisted of analysing existing data from NEB (Galway pilot) and VTEC sensors in order to identify a general entity relationship model for modelling sensors and their data. This work was carried out in WP1 and details about the entity relationship model are available in D1.3. This appendix is concerned about phase 2 of this process that consists of (1) analysing the existing entity relationship model provided in Deliverable D1.3, (2) creating a direct mapping from the existing data model to RDF concepts (classes and properties in RDF) and finally (3) determine a mapping/matching to existing ontological concepts resulting into a second version of the minimal vocabulary.

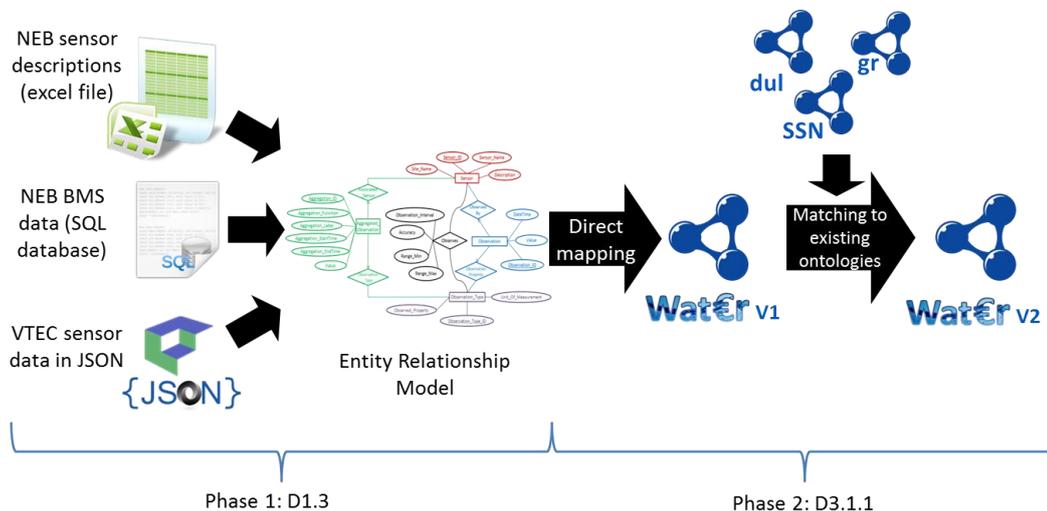


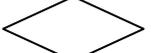
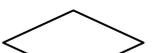
Figure 48 – Phase 1 and 2 of the design of the RDF minimal vocabulary for WatErnomics

- Direct mapping:** A direct mapping consists of identifying the set of classes and properties required to model the concepts of the entity relationship model taken from D1.3. We propose the mapping proposed in Table 8 to translate entity relationship model concepts into RDF. This work output is a first version of the minimal vocabulary. Details of this mapping will follow in this appendix.
- Matching to existing ontologies:** We reuse existing vocabularies that are proven to be effective and compliant with most of the sensor modelling standards. Indeed, we focus mainly on reusing concepts from Semantic Sensor Network (SSN) ontology [17] for modelling sensors and their observations. In fact, SSN is built while being compliant to the W3C and the Open Geospatial Consortium (OGC) [18] standard SensorML [19]. In addition to SSN, other ontologies were also considered such as *gr*¹ and *dul*². The output of this work is a second version of the minimal vocabulary. Details of this matching will follow in this appendix.

¹ *gr* stands for good relations ontology available at <http://purl.org/goodrelations/v1#>

² *dul* stands for DOLCE+DnS Ultralite ontology available at <http://www.loa-cnr.it/ontologies/DUL.owl#>

Table 8: Mapping from entity relationship item to RDF concepts

Item	Graphical representation	RDF type
Entity	Rectangle 	rdf:Class
Attribute	Oval 	rdf:Property
Relationship without attributes	Diamond 	rdf:Property
Relationship with attributes	Diamond 	rdf:Class

Designing the minimal vocabulary for the Waternomics project is a continuous work that will be carried out along with WP3 tasks. First, other pilot sites’ restrictions and models and sensors should be covered by this vocabulary. Second, the development of support services and applications might require additional data sources such as new entities, new sensors and other open data. This leads to **phase 3** (see Figure 49) of the design of the Waternomics minimal vocabulary. Similar to **phase 2** (see Figure 48), two steps are required (1) identification of relevant ontological concepts and including them to the minimal vocabulary then (2) matching the new concepts to existing ontologies. The final output of these updates will be included in D3.1.2.

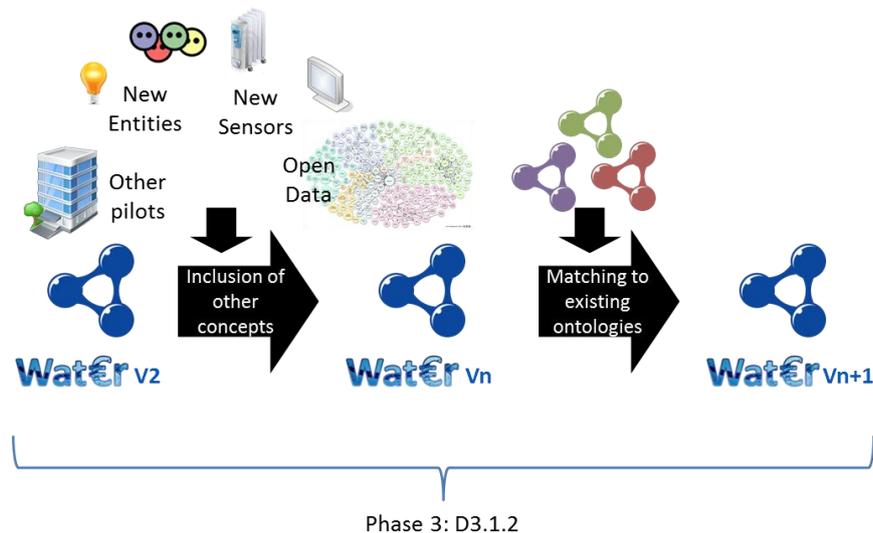


Figure 49 – Phase 3 of the design of the RDF minimal vocabulary for Waternomics

In the rest of the document, we use *water*¹ as a prefix to refer to the water vocabulary that contains all the classes and properties generated from the entity relationship model.

¹ @prefix water:<<http://waternomics.eu/ontology/water#>>.

Please note that throughout this appendix we use an online RDF translator¹ to validate the syntax of the examples proposed. We also use the W3C validation Service² for creating the corresponding RDF graphs.

2. Sensor to RDF

2.2 Entity relationship to RDF mapping

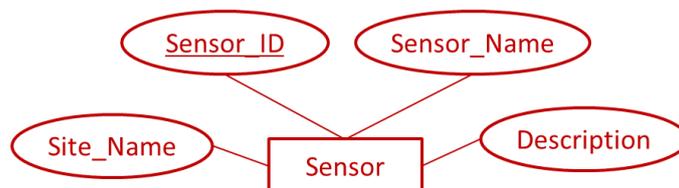


Figure 50 – Entity Relationship Model: Sensor

Table 9: Mapping Sensor Entity Relationship Model to RDF

Item	RDF Concept	Mapping to an existing ontology concept
Sensor	water:Sensor (rdf:Class)	ssn:Sensor
Sensor_ID	water:sensor_ID (rdf:Property)	
Sensor_Name	water:sensorName (rdf:Property)	rdfs:label
Description	water:description (rdf:Property)	rdfs:comment
Site_Name	water:site_Name (rdf:Property)	dul:hasLocation

2.3 Example

In our example we will use the following base URI : <http://waternomics.eu/data/>

Table 10: Sensor Table Example

Sensor_ID	Sensor_Name	Site_Name	Description
1	Vtech Sensor	VTEC	This is VTEC building
2	NUIG Sensor	NUIG	This sensor is installed in the NUIG Engineering Building

Listing 3: Sensor Example RDF n3

```

@base <http://waternomics.eu/data/>.
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix dul: <http://www.loa-cnr.it/ontologies/DUL.owl#>.
@prefix water: <http://waternomics.eu/ontology/water#>.

<sensor/sensor_ID=1> a ssn:Sensor.
<sensor/sensor_ID=1>water:sensor_ID "1"^^xsd:string.
  
```

¹<http://rdf-translator.appspot.com/>

²<http://www.w3.org/RDF/Validator/>

```

<sensor/sensor_ID=1>rdfs:label      "Vtech Sensor"^^xsd:string.
<sensor/sensor_ID=1>rdfs:comment   "This is VTEC building"^^xsd:string.
<sensor/sensor_ID=1>dul:hasLocation "Vtech"^^xsd:string.

<sensor/sensor_ID=2>      a          ssn:Sensor.
<sensor/sensor_ID=2>water:sensor_ID "2"^^xsd:string.
<sensor/sensor_ID=2>rdfs:label      "NUIG Sensor"^^xsd:string.
<sensor/sensor_ID=2>rdfs:comment   "This sensor is installed in the NUIG Engineering Building"^^xsd:string.
<sensor/sensor_ID=2>dul:hasLocation "NUIG"^^xsd:string.
    
```

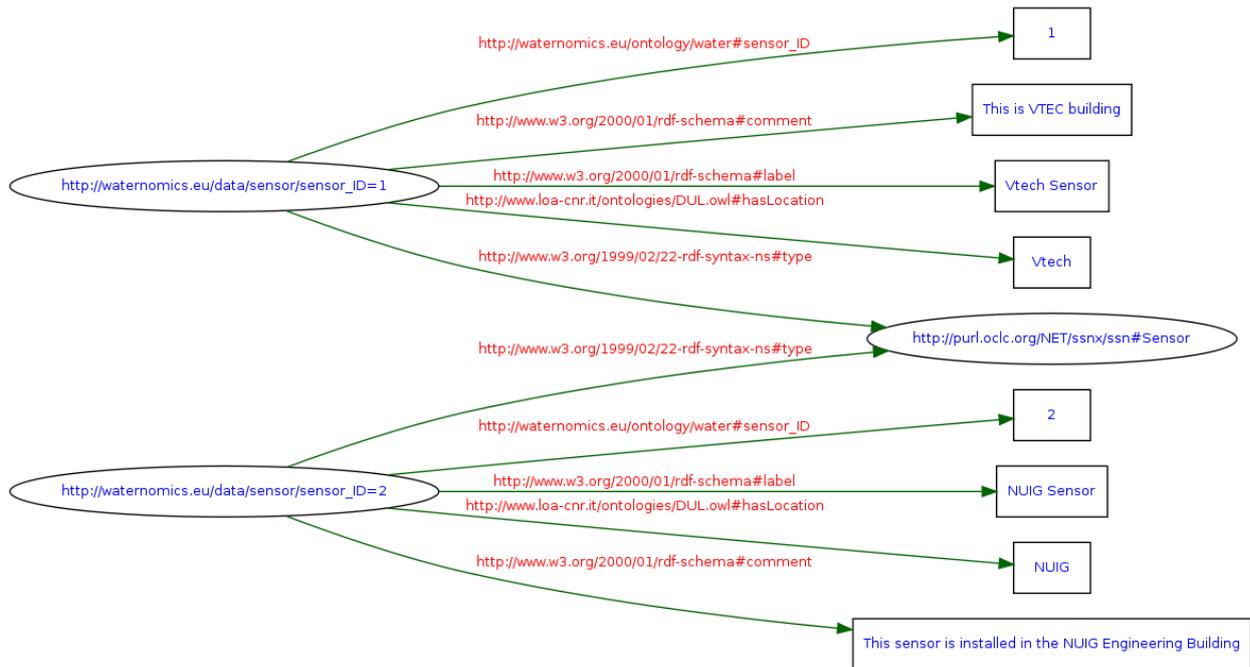


Figure 51 – RDF graph example for Sensor

3. Observation_Type to RDF

3.2 Entity relationship to RDF mapping

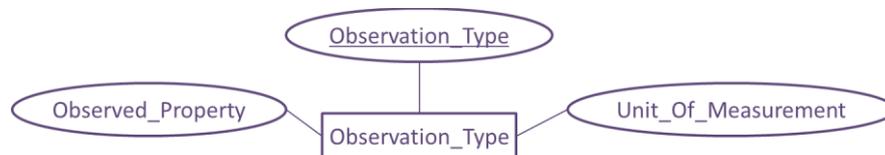


Figure 52 – Entity Relationship Model: Observation_Type

Table 11: Mapping Observation_Type Entity Relationship Model to RDF

Item	RDF Concept	Mapping to an existing ontology concept
Observation_Type	water:Observation_Type (rdf:Class)	ssn:Property
Observed_Property	water:observed_Property (rdf:Property)	
Observation_Type_ID	water:observation_Type_ID (rdf:Property)	
Unit_Of_Measurement	water:unit_Of_Measurement (rdf:Property)	gr:hasUnitOfMeasurement

3.3 Example

Table 12: Observation_Type Table Example

Observation_Type_ID	Observed_Property	Unit_Of_Measurement
fvel	Flow Velocity	m/s
flow	Flow	m3/h
waterc	Water Consumption	m3

Listing 4: Observation_Type Example RDF n3

```

@base <http://waternomics.eu/data/>.
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix gr: <http://purl.org/goodrelations/v1#>.
@prefix water:<http://waternomics.eu/ontology/water#>.

<observation_Type/observation_Type_ID=fvel> a
                                                ssn:Property.
<observation_Type/observation_Type_ID=fvel>water:observationType_ID "fvel"^^xsd:string.
<observation_Type/observation_Type_ID=fvel>water:observed_Property "Flow Velocity"^^xsd:string.
<observation_Type/observation_Type_ID=fvel>gr:hasUnitOfMeasurement "m/s"^^xsd:string.

<observation_Type/observation_Type_ID=flow> a
                                                ssn:Property.
<observation_Type/observation_Type_ID=flow>water:observationType_ID "flow"^^xsd:string.
<observation_Type/observation_Type_ID=flow>water:observed_Property "Flow"^^xsd:string.
<observation_Type/observation_Type_ID=flow>gr:hasUnitOfMeasurement "m3/s"^^xsd:string.

<observation_Type/observation_Type_ID=waterc> a
                                                ssn:Property.
<observation_Type/observation_Type_ID=waterc>water:observationType_ID "waterc"^^xsd:string.
<observation_Type/observation_Type_ID=waterc>water:observed_Property "Water Consumption"^^xsd:string.
<observation_Type/observation_Type_ID=waterc>gr:hasUnitOfMeasurement "m3"^^xsd:string.

```



Figure 53 – RDF graph example for Observation_Type

4. Sensor_Observation_Type to RDF

4.2 Entity relationship to RDF mapping

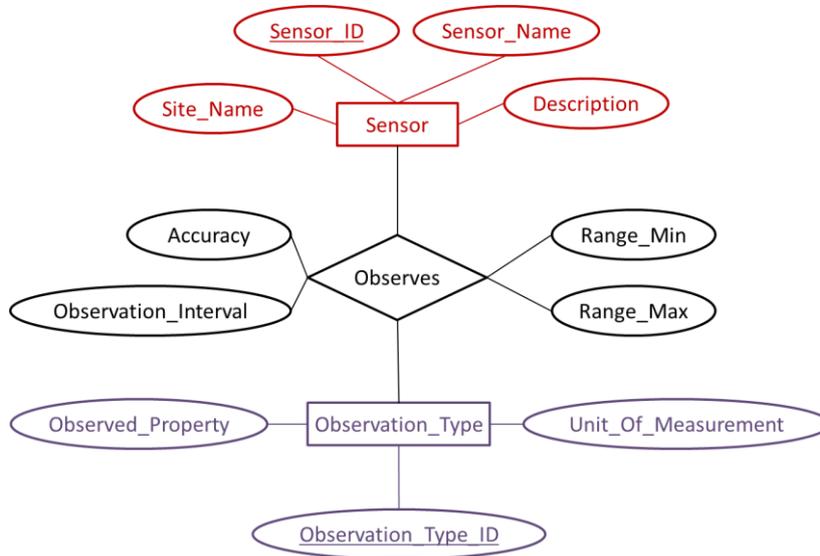


Figure 54 – Entity Relationship Model: Sensor_Observation_Type captured as Observes relation

Table 13: Mapping Sensor_Observation_Type Entity Relationship Model to RDF

Item	RDF Concept	Mapping to an existing ontology concept
Observes (Sensor_Observation_Type)	water:Sensor_Observation_Type (rdf:Class)	ssn:MeasurementProperty
Sensor_ID	water:associated_Sensor	

Observation_Type_ID	water:associated_Observation_Type	
Observation_Interval	water:observation_Interval (rdf:Property)	
Accuracy	water:accuracy (rdf:Property)	
Range_Min	water:range_Min (rdf:Property)	gr:hasMaxValue
Range_Max	water:range_Max (rdf:Property)	gr:hasMinValue

4.3 Example

Table 14: Sensor_Observation_Type Table Example

Sensor_ID	Observation_Type_ID	Range_Min	Range_Max	Accuracy	Observation_Interval
1	fvel	0	200	10	20
1	flow	0	200	1	20

Listing 5: Sensor_Observation_Type Example RDF n3

```

@base <http://waternomics.eu/data/>.
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix gr: <http://purl.org/goodrelations/v1#>.
@prefix water: <http://waternomics.eu/ontology/water#>.

<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=fvel> a ssn:MeasurementProperty.
<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=fvel>gr:hasMinValue"0"^^xsd:double.
<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=fvel>gr:hasMaxValue"200"^^xsd:double.
<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=fvel>water:accuracy"10"^^xsd:double.
<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=fvel>water:observation_Interval"20"^^xsd:integer.
<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=fvel>water:associated_Sensor<sensor/sensor_ID=1
>.
<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=fvel>water:associated_Observation_Type<observati
on_Type/observation_Type_ID=fvel>.

<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=flow> a ssn:MeasurementProperty.
<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=flow>gr:hasMinValue"0"^^xsd:double.
<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=flow>gr:hasMaxValue"200"^^xsd:double.
<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=flow>water:accuracy"1"^^xsd:double.
<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=flow>water:observation_Interval"20"^^xsd:integer.
<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=flow>water:associated_Sensor<sensor/sensor_ID=1
>.
<sensor_Observation_Type/sensor_ID=1,observation_Type_ID=flow>water:associated_Observation_Type<observa
tion_Type/observation_Type_ID=flow>.

```



Figure 55 – RDF graph example for `Sensor_Observation_Type`

5. Observation to RDF

5.2 Entity relationship to RDF mapping

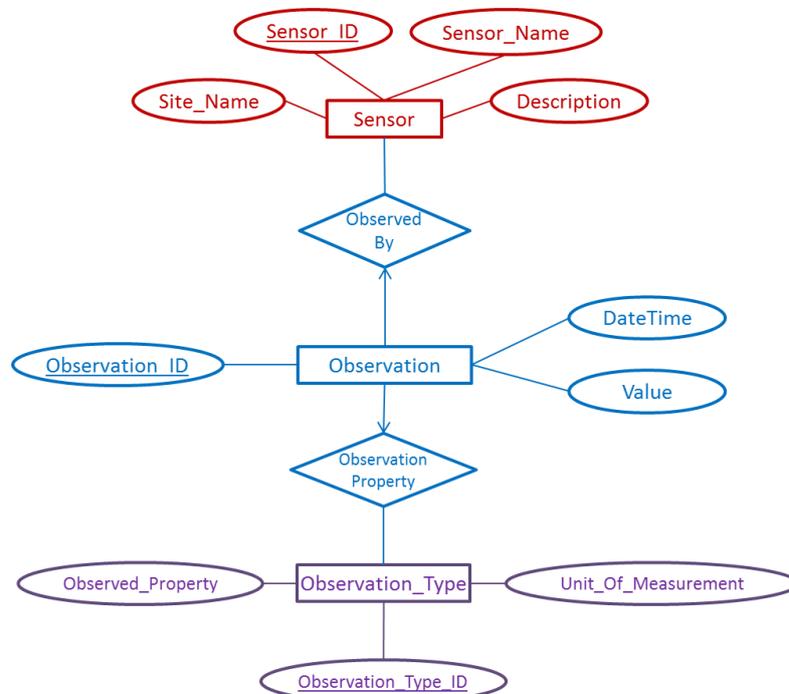


Figure 56 – Entity Relationship Model: Observation

Table 15: Mapping Observation Entity Relationship Model to RDF

Item	RDF Concept	Mapping to an existing ontology concept
Observation	water:Observation (rdf:Class)	ssn:Observation
Observation_ID	water:observation_ID(rdf:Property)	
DateTime	water:dateTime (rdf:Property)	
Value	water:value (rdf:Property)	owl:hasValue
Observed_By	water:observed_By (rdf:Property)	ssn:observedBy
Observation_Property	water:observation_Property (rdf:Property)	

5.3 Example

Table 16: Observation Table Example

Observation_ID	Sensor_ID	Observation_Type_ID	DateTime	Value
0	1	fvel	1417786723000	1429.6
1	1	flow	1417786723000	1.32423

Listing 6: Observation Example RDF n3

```

@base <http://waternomics.eu/data/>.
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix gr: <http://purl.org/goodrelations/v1#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix water:<http://waternomics.eu/ontology/water#>.

<observation/observation_ID=0> a ssn:Observation.
<observation/observation_ID=0>water:observation_ID"0"^^xsd:string .
<observation/observation_ID=0>ssn:observedBy<sensor/sensor_ID=1>.
<observation/observation_ID=0>water:observation_Property<observation_Type/observation_Type_ID=fvel>.
<observation/observation_ID=0>water:dateTime"1417786723000"^^xsd:double .
<observation/observation_ID=0>owl:hasValue"1429.6"^^xsd:double .

<observation/observation_ID=1> a ssn:Observation.
<observation/observation_ID=1>water:observation_ID"1"^^xsd:string .
<observation/observation_ID=1>ssn:observedBy<sensor/sensor_ID=1>.
<observation/observation_ID=1>water:observation_Property<observation_Type/observation_Type_ID=flow>.
<observation/observation_ID=1>water:dateTime"1417786723000"^^xsd:double .
<observation/observation_ID=1>owl:hasValue"1.32423"^^xsd:double .
    
```

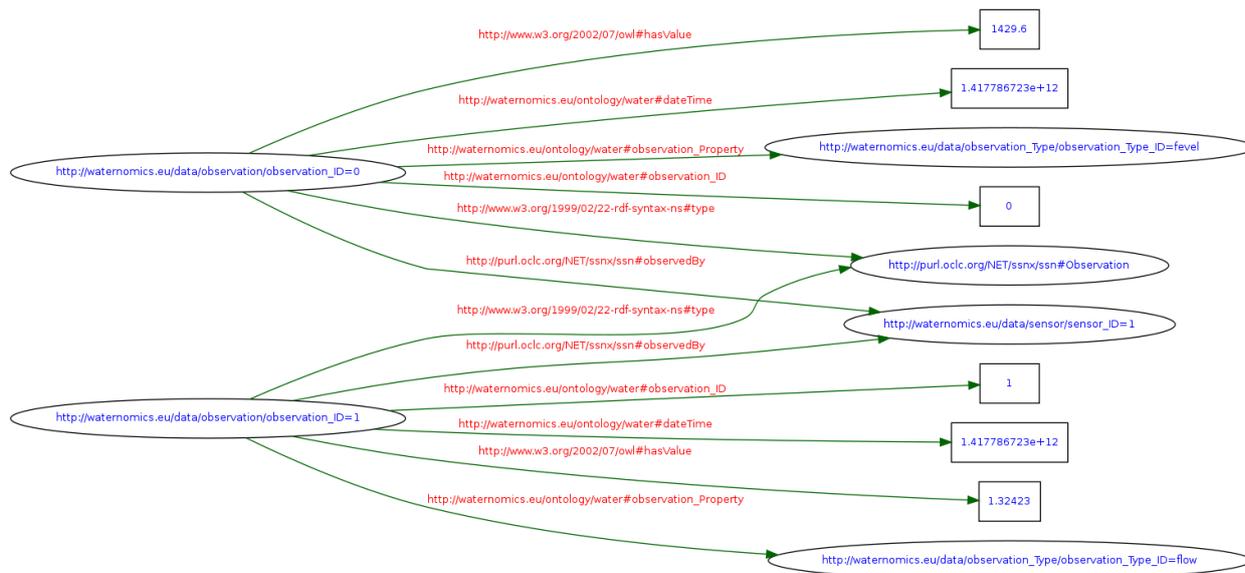


Figure 57 – RDF graph example for Observation

6. Observation to RDF

6.1 Entity relationship to RDF mapping

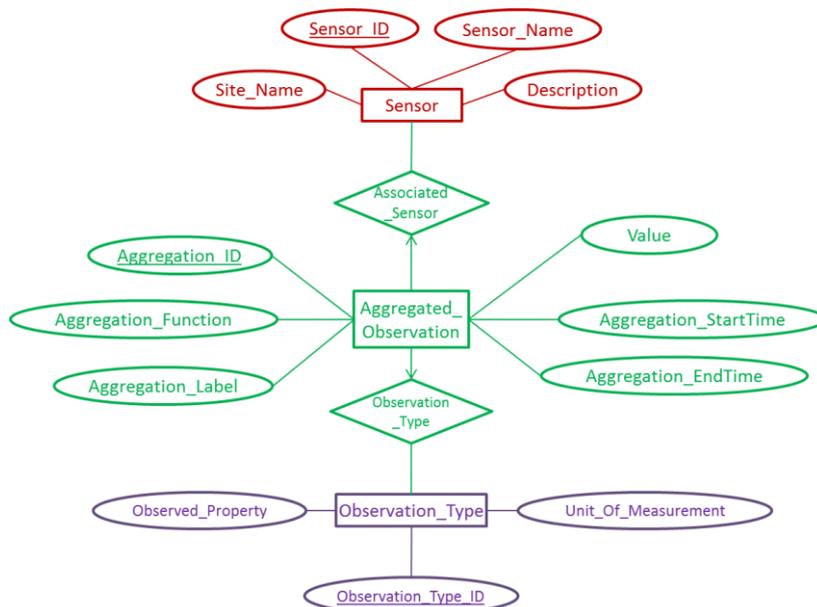


Figure 58 –Entity Relationship Model: Aggregated_Observation

Table 17: Mapping Aggregated_Observation Entity Relationship Model to RDF

Item	RDF Concept	Mapping to an existing ontology concept
Aggregated_Observation	water:Aggregated_Observation (rdf:Class)	ssn:Observation
Aggregation_ID	water:aggregation_ID(rdf:Property)	
Aggregation_Label	water:aggregation_Label (rdf:Property)	rdfs:label

Aggregation_Function	water:aggregation_Function (rdf:Property)	
Aggregation_StartTime	water:aggregation_StartTime (rdf:Property)	
Aggregation_EndTime	water:aggregation_EndTime (rdf:Property)	
Value	water:value (rdf:Property)	owl:hasValue
Sensor_ID	water:assotiated_Sensor (rdf:Property)	
Observation_Type_ID	water:associated_Observation_Type (rdf:Property)	

6.2 Example

Table 18: Aggregated_Observation Table Example

Aggregation_ID	Sensor_ID	Observation_Type_ID	Aggregation_Function	Aggregation_Label	Aggregation_StartTime	Aggregation_EndTime	Value
123	1	waterc	sum	today	141773760000	1417786723000	7.1783

Listing 7: Aggregated_Observation Example RDF n3

```

@base <http://waternomics.eu/data/>.
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix water: <http://waternomics.eu/ontology/water#>.

<aggregated_observation/aggregation_ID=123> a ssn:Observation.
<aggregated_observation/aggregation_ID=123>water:aggregation_ID"123"^^xsd:string .
<aggregated_observation/aggregation_ID=123>water:associated_Sensor<sensor/sensor_ID=1>.
<aggregated_observation/aggregation_ID=123>rdfs:label"today"^^xsd:string .
<aggregated_observation/aggregation_ID=123>water:aggregation_Function"sum"^^xsd:string .
<aggregated_observation/aggregation_ID=123>water:associated_Observation_Type<observation_Type/observation_Type_ID=waterc>.
<aggregated_observation/aggregation_ID=123>water:aggregation_StartTime"141773760000"^^xsd:double .
<aggregated_observation/aggregation_ID=123>water:aggregation_EndTime"1417786723000"^^xsd:double .
<aggregated_observation/aggregation_ID=123>owl:hasValue"7.1783"^^xsd:double .
    
```

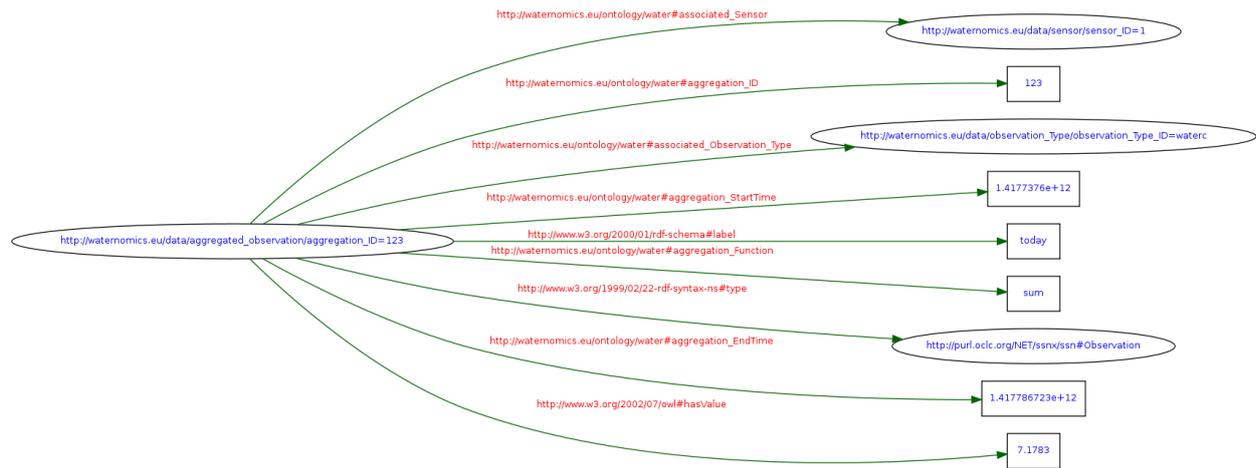


Figure 59 – RDF graph example for *Aggregated_Observation*

Appendix B: Survey of Open Data for Water Management

As a result of the development in the field of computing (i.e. data storage and so forth) and new devices, the possibility to acquire (and consequently store) data grew considerably in the last years. In fact, nowadays, many organizations, companies and agencies have the possibility to collect data as never before in the history of the human being[46]. In particular, within the goal of Waternomics (water saving), data on water (e.g. water consumption, leakages, flow rate and pressure in water network) are collected and “available” in a publishing platform. This is a big deal for several stakeholders and researchers as the possibility to store new kind of data can generate new opportunities of revenues in both research and consulting services. As an example of opportunities, these sets of data allow researchers to track and observe water across scales and timeframes, making it possible to reconstruct and understand the end users water consumption processes, how to engage users, stimulate water usage efficiency, share water information at time – scale relevant for decision making and helpful to increase end-users awareness to converge towards an eco-system of responsible water usage.

Another important task within the management of the water data set comes from the possibility to acquire and analyse data from satellites. This data can provide as an example:

- information on groundwater levels;
- assessment related to water quality across large areas.

For what the first point is concerned, the use of the related data may allow the researchers to identify potential challenges for people and ecosystems. On the other hand, through the second point, satellite data may identify changes in water quality more quickly with respect the on-the-ground sampling, with a great benefit for humans’ health through the insurance of continuous and near real time monitoring.

Thus, the collection and analysis of data about water, besides helping the connection between researchers and governments with individual water users, also raises awareness of water management challenges and supports transparency in water governance.

An example about the effectiveness of the data sets is represented by the Global Earth Observation System of Systems (GEOSS)[47]. It will be a global and flexible network of content providers allowing decision makers to access an extraordinary range of information at their desk, simply simultaneously addressing nine areas of critical importance to people and society (see Figure 60). This system links together existing and planned observing systems around the world and will support the development of new systems where gaps currently exist. The related ‘GEOSS Portal’ (see Figure 61) offers a point for users looking for data (both with or without Internet access), imagery and analytical software packages relevant to all parts of the globe, simply connecting users to existing data bases and portals and provides reliable, up-to-date and user friendly information – vital for the work of decision makers, planners and emergency managers.



Figure 60: GEOSS logo[47]



Figure 61: GEOSS Portal [47]

Nowadays many government agencies all around the world are promoting and encouraging the development of open data about water resources and related aspects; e.g.:

- AfriGEOSS program as part of the GEOSS project. It is expected to help African agencies respond to natural disasters such as floods and forest fires by supporting the direct download of satellite data;
- US Geological Survey;
- The WISDOM Project [17];
- Kenya Open Data Initiative.

On the one hand, the above approaches related to the creation of a large number of data sets, offers a potential tool for both research and evidence-based management of water resources; on the other hand, it creates some issue related to the accessibility of the data itself. In fact, these potentialities can be fully realized only if data is available and accessible in a form that supports their use. That means that data has to be published following a certain structure or schema. Basically the challenge is to store the data in a format that can be easily read and understood by other ICT tools. This is can be insured using the Semantic Web best practice in order to define and use proper ontologies and publish the data following those schemas, thus producing Linked Open Data.

These aspects are summarized in the “Data Sharing Principles” established by GEOSS[48]. In brief, these principles show special attention to:

- metadata: descriptive information about data that supports their use;
- data interoperability: a functional level of consistency that allows systems to work together.

Relatively to these aspects, a more difficult and delicate challenge is posed by legal restrictions and data ownership. It may vary across countries – and sometimes even within the same country some conflict or issue may arise. Moreover, the data collection sometimes is really expensive from the economic point of view and time consuming. Thus, some agencies and/or companies that collect data need to sell these collected data sets to partially or totally cover their costs. In other words, this means that only some of the published data are freely and openly available for the end users. These “restrictions” constitute a real barrier for researchers and stakeholders, limiting knowledge-sharing and potentially hindering the development of tools for water users.

Within the scientific community the attitudes toward data sharing may vary widely, because, as already observed, collecting and organizing environmental monitoring data or observations can take a lot of time and effort. As one countermeasure to protect data but at the same time to incentivize their production, scientific journals are establishing policies for data access, supporting data storage and access and even providing dedicated venues for data publication, such as Earth System Science Data(see [49] for more details).

In Europe, a consolidated policy recommendations for open access to research data are being developed within the project RECODE (Figure 62Figure 62); see [50] for more details.



Figure 62: RECODE logo

Another correlated problem is represented by the way this data are presented to the user. It is a challenging problem to present advanced GUI to the end-user, and a proper user-friendly way to show linked data should be individuated and drawn[51]. Tools that can be used for appealing linked data visualization are proposed in the literature[52][53][54].Figure 63 shows the look and feel of [53] that can be used for any ontology.

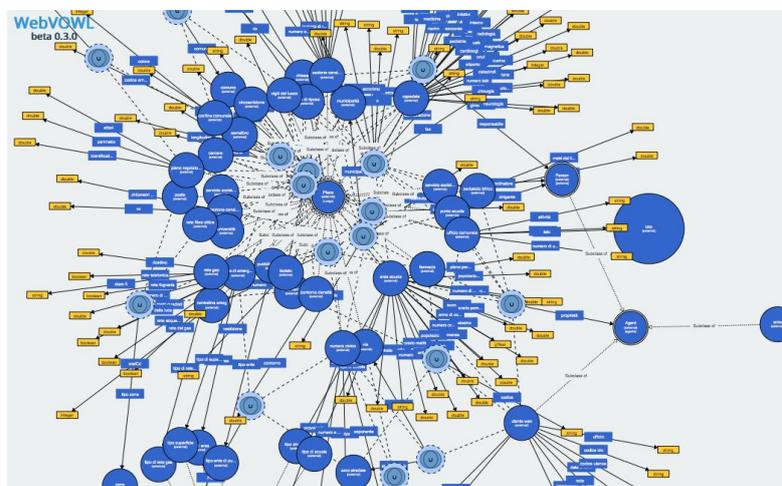


Figure 63: Look and feel of Vowl.

Examples of published open data where semantic web best practices have been designed on top are:

- U.S. Geological Survey real-time flow data: <http://waterdata.usgs.gov/nwis/rt>
- INSPIRE: <http://inspire.ec.europa.eu/index.cfm>
- EU Open Data Portal: <http://open-data.europa.eu/en/data/>
- Integrated Earth Data Applications (IEDA): <http://www.iedadata.org/>
- Kenya Open Data: <https://opendata.go.ke>
- Open water foundation: <http://openwaterfoundation.org/>
- Water Data catalog – city of Toronto: <http://www1.toronto.ca/wps/portal/contentonly?vgnextoid=8517e03bb8d1e310VgnVCM10000071d60f89RCRD>
- Open water data initiative: http://acwi.gov/acwi-minutes/acwi2014/Open_water_data_proposal_to_acwi-7-21-14.pdf
- UK WaterAid: <http://www.wateraid.org/uk/who-we-are/open-data>

One more example is constituted by the WISDOM Project[55], a bilateral FP7 research project started in 2007 between Germany and Vietnam whose focus is the creation of a Water related Information System for the Mekong Delta (Figure 64 shows a screenshot of the project home page). More in detail, the main objective is the collection of data on the Mekong Delta including several fields such as Hydrology, Geography, Earth Observation, and Sociology. This data is then integrated into a single Information System. The rationale behind that is having a system able to combine information of different factors and provide a decision support system to questions such as: in case of a flood, how much agricultural lands have been affected? Ontologies describing spatial, thematic, and temporal reference aspects have been implemented in order to provide semantic information that can be added to the datasets. This also allows a quicker data retrieval by meaningful search attributes as e.g. look for all data in a certain administrative area with respect to a certain theme. The WISDOM project aims to save water and energy through the integration of innovative Information and Communication Technologies (ICT) frameworks to optimize water distribution networks and to enable change in consumer behaviour through innovative demand management and adaptive pricing schemes. To do that, the approach used within the WISDOM project makes use of sensor monitoring and communication systems with semantic modelling and control capabilities to provide for near real-time management of urban water resources.

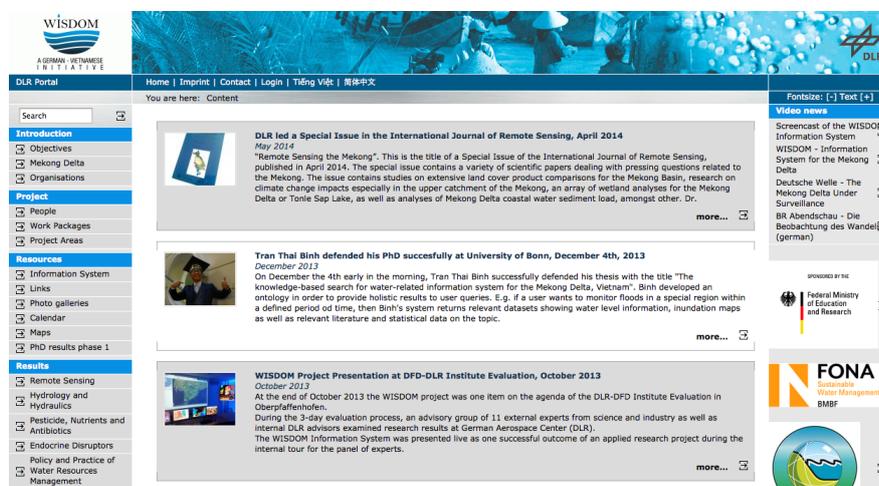


Figure 64: Screenshot of the WISDOM project website

As far as the ICT tools are concerned, an existing solution that can be used for open data management within the water saving domain is constituted by the sMAP project ([56], [57]). It provides:

- (i) a specification for transmitting physical data and describing its contents,
- (ii) a wide choice of open drivers to communicate with devices using native protocols and
- (iii) everything needed to query large repository of physical data.

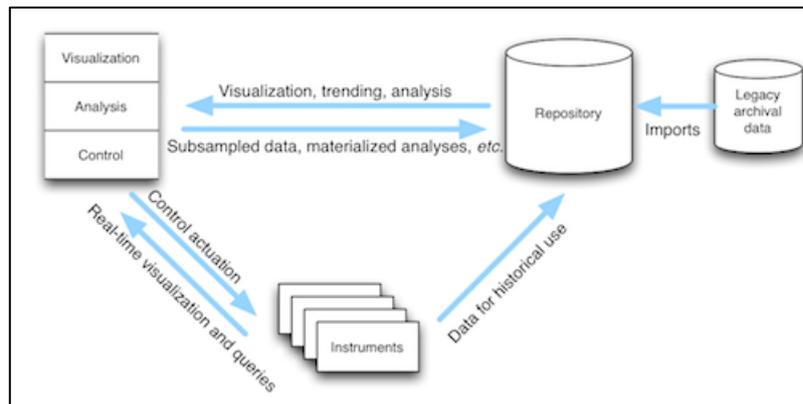


Figure 65: Architecture of the sMAP project¹

Figure 65 shows the architecture of the sMAP, which consists of several components adopted to capture, transmit, store time series data.

sMAP Sources collect time series data using different protocols (such as Modbus, BACnet, or HTTP/XML) using sMAP library to communicate the data to consumers.

The **sMAP Archiver** provides an interface to store data and use real time data, which adopts the JSON format in order to represent it. It makes use of SQL language and REST API to select time series.

Applications provide appealing visualizations for time series, compute control strategies and provide user feedback.

The basic way to use sMAP is to create and configure data sources, which feed into the overall system. The sMAP source library is developed in Python programming language and makes use of *twisted*[78], an asynchronous event system which serves sMAP resources over HTTP and manages the forward of data. The sMAP archiver is a separate script developed in Python as well. It is built on top of *readingdb*[59], an optimized time series database.

Pachube [60] is a hosted web application used for collecting and cataloging a large variety of sensor data with a HTTP/JSON interface similar to sMAP. It enables people to monitor and share real time environmental data from sensors that are connected to the Internet. It works between environments and it is able to capture input data from remote sensors and serve output data to remote actuators. It can be used in physical environments and it also allows people to embed sensor data in web-pages. After the loading of the data into the system, different visualization and analytics plugins are available to examine readings. The system might be considered as an example of the type of universal application which can be built on top of sMAP. It has been acquired by LogMeIn in July 2011 and now known as Xively.com. Figure 66 shows the old website where Pachube was hosted.

¹<http://www.cs.berkeley.edu/~stevedh/smap2/>



Figure 66: Pachube old website

Researchers in [61] have shown a use case related to integrated water resources on how to connect data sources available on the Web using Linked Data. In specific they have formalized the Integrated Water Resources Management (IWRM) knowledge and decision support domain in an OWL ontology, which reuses the RDF Data Cube Vocabulary[62]. Linked Data has been used to represent, extract, integrate and load community-created research and sensor data into a knowledge base where data can be queried and browsed. Last but not least, the authors presented consumption tools on top of the IWRM knowledge base allowing scientist to share and re-use research data. The work has been carried out within the SMART II project[63]. Figure 67 shows the architecture of the SMART Knowledge Base (SKB), which is divided into the Data Sources and the Data Consumption Tools. A triple store allows the consumption tools to access data from the data sources. Data is then further processed and shown to the users. Data sources consist of:

- (i) an IWRM ontology, developed in
- (ii) Dropedia[64], collaborative knowledge management system of the SMART project based on the Open Source semantic wiki software, Semantic MediaWiki and
- (iii) the SMART-DB, where data is published as Linked Data for integration in the SMART Knowledge Base using a Google-App-Engine which translates XML into an RDF representation using the IWRM ontology.

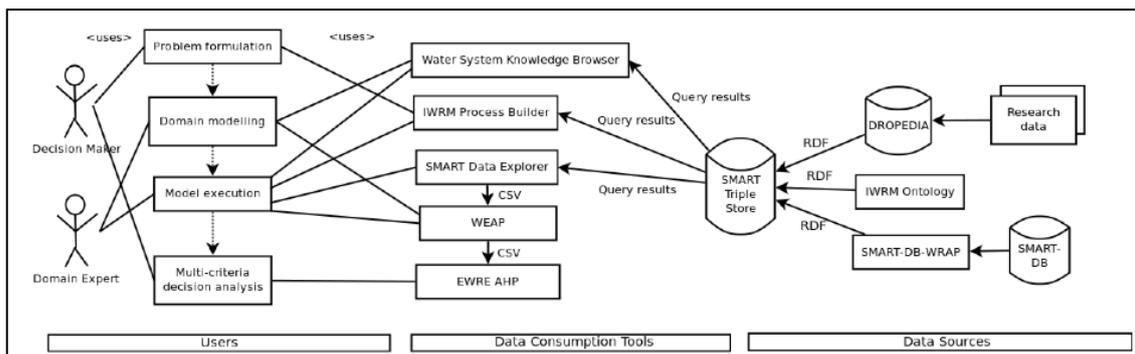


Figure 67: Architecture of the SMART Knowledge Base [61]

8 References

- [1] "The Open Definition." [Online]. Available: <http://opendefinition.org/>.
- [2] E. Bruke, "An Autonomic Approach to Real-Time Predictive Analytics using Open Data and the Web of Things," National University of Ireland, Galway, 2013.
- [3] T. Berners-Lee, "Linked Data- Design Issues," 2006. [Online]. Available: <http://www.w3.org/DesignIssues/LinkedData.html>. [Accessed: 12-May-2013].
- [4] CERN, "The birth of the Web." .
- [5] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," 2005. [Online]. Available: <http://merlot.tools.ietf.org/html/rfc3986>. [Accessed: 01-Mar-2012].
- [6] T. Berners-Lee and D. Connolly, "Hypertext markup language-2.0," 1995.
- [7] T. Berners-Lee, T. Bray, D. Connolly, P. Cotton, R. Fielding, M. Jeckle, C. Lilley, N. Mendelsohn, D. Orchard, N. Walsh, and others, "Architecture of the world wide web, volume one," *version*, 2004. [Online]. Available: <http://www.w3.org/TR/2004/REC-webarch-20041215/>.
- [8] T. B. Lee, J. Hendler, and O. Lassila, "The semantic web," *Sci. Am.*, vol. 284, no. 5, pp. 34–43, 2001.
- [9] W3C, "Semantic Web Layer Cake," 2007. [Online]. Available: <http://www.w3.org/2007/03/layerCake.png>.
- [10] G. Klyne and J. J. Carroll, "Resource Description Framework (RDF): Concepts and Abstract Syntax," 2004. [Online]. Available: <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>. [Accessed: 25-Feb-2013].
- [11] T. and others Bizer, Chris and Cyganiak, Richard and Heath, "How to Publish Linked Data on the Web," 2007. [Online]. Available: <http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/LinkedDataTutorial/>.
- [12] D. Brickley and R. V Guha, "{RDF vocabulary description language 1.0: RDF schema}," 2004.
- [13] D. L. McGuinness, F. Van Harmelen, and others, "OWL web ontology language overview," *W3C Recomm.*, vol. 10, pp. 2003–2004, 2004.
- [14] E. Prud'Hommeaux and A. Seaborne, "SPARQL query language for RDF," *W3C Work. Draft*, vol. 4, no. January, 2008.
- [15] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A Nucleus for a Web of Open Data," *Semant. Web*, vol. 4825, pp. 722–735, 2007.
- [16] R. C. Max Schmachtenberg, Christian Bizer, Anja Jentzsch, "Linking Open Data cloud diagram 2014," 2014. [Online]. Available: <http://lod-cloud.net/>.
- [17] M. Compton, P. Barnaghi, L. Bermudez, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, and A. Sheth, "The SSN Ontology of the W3C Semantic Sensor Network Incubator Group," 2011.
- [18] "OGC - Open Geospatial Consortium." .
- [19] "SensorML - Sensor Model Language." .
- [20] S. Das, S. Sundara, and R. Cyganiak, "R2RML: RDB to RDF Mapping Language (W3C Recommendation 27 September 2012)." 2011.
- [21] J. F. Sequeda and D. P. Miranker, "Ultrawrap: SPARQL Execution on Relational Data," *Web Semant.*, vol. 22, pp. 19–39, 2013.
- [22] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. de Walle, "{RML}: A Generic Language for Integrated {RDF} Mappings of Heterogeneous Data," in *Proceedings of the 7th Workshop on Linked Data on the Web*, 2014.
- [23] J. Golbeck, M. Grove, B. Parsia, A. Kalyanpur, and J. A. Hendler, "New Tools for the Semantic Web," in *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, 2002, pp. 392–400.
- [24] T. Lebo and G. T. Williams, "Converting governmental datasets into linked data," in *I-SEMANTICS*, 2010.
- [25] L. Han, T. Finin, C. Parr, J. Sachs, and A. Joshi, "RDF123: From Spreadsheets to RDF," in *The Semantic Web - ISWC 2008*, 2008, vol. 5318, pp. 451–466.
- [26] A. Langegger and W. WoB, "XLWrap - Querying and Integrating Arbitrary Spreadsheets with SPARQL," in *The Semantic Web - ISWC 2009*, 2009, vol. 5823, pp. 359–374.
- [27] D. Loshin, *Master Data Management*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008.
- [28] A. Berson and L. Dubov, *Master Data Management and Customer Data Integration for a Global Enterprise*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2007.

- [29] X. Wang, X. Sun, F. Cao, L. Ma, N. Kanellos, K. Zhang, Y. Pan, and Y. Yu, "SMDM : Enhancing Enterprise-Wide Master Data Management Using Semantic Web Technologies," *Proc. VLDB Endow.*, vol. 2, no. 2, pp. 1594–1597, 2009.
- [30] B. Otto and A. Reichert, "Organizing Master Data Management: Findings from an Expert Survey," in *Proceedings of the 2010 ACM Symposium on Applied Computing - SAC '10*, 2010, pp. 106–110.
- [31] B. Otto, "Data Governance," *Bus. Inf. Syst. Eng.*, vol. 3, no. 4, pp. 241–244, Jun. 2011.
- [32] L. Haas, "Beauty and the Beast: The Theory and Practice of Information Integration," in *Database Theory – ICDT 2007*, vol. 4353, T. Schwentick and D. Suciu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 28–43.
- [33] A. Haug and J. S. Arlbjørn, "Barriers to master data quality," *J. Enterp. Inf. Manag.*, vol. 24, no. 3, pp. 288–303, 2011.
- [34] P. A. Bernstein and L. M. Haas, "Information integration in the enterprise," *Commun. ACM*, vol. 51, no. 9, p. 72, Sep. 2008.
- [35] A. Doan, R. Ramakrishnan, and A. Y. Halevy, "Crowdsourcing systems on the World-Wide Web," *Commun. ACM*, vol. 54, no. 4, p. 86, Apr. 2011.
- [36] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, *Item-based collaborative filtering recommendation algorithms*. New York, New York, USA: ACM Press, 2001, pp. 285–295.
- [37] M. A. Suryanto, E. P. Lim, A. Sun, and R. H. L. Chiang, "Quality-aware collaborative question answering," in *Proceedings of the Second ACM International Conference on Web Search and Data Mining - WSDM '09*, 2009, pp. 142–151.
- [38] J. Bian, Y. Liu, E. Agichtein, and H. Zha, "Finding the right facts in the crowd: factoid question answering over social media," in *Proceeding of the 17th international conference on World Wide Web - WWW '08*, 2008, pp. 467–476.
- [39] K. K. Nam, M. S. Ackerman, and L. A. Adamic, "Questions in, knowledge in?: a study of naver's question answering community," in *Proceedings of the 27th international conference on Human factors in computing systems - CHI '09*, 2009, pp. 779–788.
- [40] J. Swarts, "The collaborative construction of 'fact' on Wikipedia," in *Proceedings of the 27th ACM international conference on Design of communication - SIGDOC '09*, 2009, pp. 281–288.
- [41] A. Kittur, E. Chi, B. A. Pendleton, and T. Mytkowicz, "Power of the Few vs . Wisdom of the Crowd : Wikipedia and the Rise of the Bourgeoisie," *Algorithmica*, pp. 1–9.
- [42] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD '08*, 2008, pp. 1247–1250.
- [43] M. Magrane and U. Consortium, "UniProt Knowledgebase: a hub of integrated protein data," *Database*, vol. 2011, Mar. 2011.
- [44] T. Palpanas, J. Chaudhry, P. Andritsos, and Y. Velegrakis, "Entity Data Management in OKKAM," in *2008 19th International Conference on Database and Expert Systems Applications*, 2008, pp. 729–733.
- [45] K. Bollacker, P. Tufts, T. Pierce, and R. Cook, "A Platform for Scalable, Collaborative, Structured Information Integration," in *Sixth International Workshop on Information Integration on the Web, AAAI-07*, 2007.
- [46] S. Kochhar, S. Mazzocchi, and P. Paritosh, "The anatomy of a large-scale human computation engine," in *Proceedings of the ACM SIGKDD Workshop on Human Computation*, 2010, pp. 10–17.
- [47] H. Stoermer, T. Palpanas, and G. Giannakopoulos, "The Entity Name System : Enabling the Web of Entities," in *IEEE 26th International Conference on Data Engineering Workshops (ICDEW)*, 2010, 2010, pp. 227–232.
- [48] H. Stoermer and P. Bouquet, "A novel approach for entity linkage," *2009 IEEE Int. Conf. Inf. Reuse Integr.*, pp. 151–156, Aug. 2009.
- [49] A. Doan, R. Ramakrishnan, F. Chen, P. Derosé, Y. Lee, R. McCann, M. Sayyadian, and W. Shen, "Community Information Management," *IEEE Data Eng. Bull.*, vol. 29, no. 1, pp. 64–72, 2006.
- [50] P. DeRose, W. Shen, F. Chen, A. Doan, and R. Ramakrishnan, "Building Structured Web Community Portals : A Top-Down , Compositional , and Incremental Approach," in *Proceedings of the 33rd International Conference on Very Large Data Bases*, 2007, pp. 399–410.
- [51] U. U. Hassan, S. O'Riain, and E. Curry, "Towards Expertise Modelling for Routing Data Cleaning Tasks within a Community of Knowledge Workers," in *Proceedings of the 17th International Conference on Information Quality*, 2012.
- [52] U. U. Hassan, S. O'Riain, and E. Curry, "Leveraging Matching Dependencies for Guided User Feedback in Linked Data Applications," in *9th International Workshop on Information Integration on the Web (IIWeb2012)*, 2012.

- [53] U. U. Hassan, M. Bassora, A. H. Vahid, S. O’Riain, and E. Curry, “A collaborative approach for metadata management for Internet of Things: Linking micro tasks with physical objects,” in *9th International Conference Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2013, pp. 593–598.
- [54] U. U. Hassan, S. O’Riain, and E. Curry, “Effects of Expertise Assessment on the Quality of Task Routing in Human Computation,” in *Proceedings of the 2nd International Workshop on Social Media for Crowdsourcing and Human Computation*, 2013.
- [55] U. U. Hassan and E. Curry, “A capability requirements approach for predicting worker performance in crowdsourcing,” in *9th International Conference Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2013, pp. 429–437.
- [56] F. Yang, E. Tschetter, X. Léauté, N. Ray, G. Merlino, and D. Ganguli, “Druid: a real-time analytical data store,” in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data - SIGMOD ’14*, 2014, pp. 157–168.
- [57] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark: Cluster Computing with Working Sets,” in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, 2010, p. 10.
- [58] Spark, “Apatch SPARK,” 2015. [Online]. Available: <https://spark.apache.org/>.
- [59] M. Franklin, A. Halevy, and D. Maier, “From Databases to Dataspaces : A New Abstraction for Information Management,” *Data Manag.*, vol. 34, no. 4, 2005.
- [60] A. Halevy, M. Franklin, and D. Maier, “Principles of dataspace systems,” *Proc. twenty-fifth ACM SIGMOD-SIGACT-SIGART Symp. Princ. database Syst. - Pod. ’06*, pp. 1–9, 2006.
- [61] N. Marz and J. Warren, *Big Data: Principles and best practices of scalable realtime data systems*. O’Reilly Media, 2013.
- [62] S. Hasan and E. Curry, “Approximate Semantic Matching of Events for the Internet of Things,” *ACM Trans. Internet Technol.*, vol. 14, no. 1, pp. 1–23, Aug. 2014.
- [63] S. Hasan and E. Curry, “Thematic event processing,” in *Proceedings of the 15th International Middleware Conference on - Middleware ’14*, 2014, pp. 109–120.
- [64] S. Hasan and E. Curry, “Thingsonomy: Tackling Variety in Internet of Things Events,” *Internet Comput. IEEE*, vol. PP, no. 99, p. 1, 2015.
- [65] S. Hasan, S. O’Riain, and E. Curry, “Towards unified and native enrichment in event processing systems,” in *Proceedings of the 7th ACM international conference on Distributed event-based systems - DEBS ’13*, 2013, p. 171.
- [66] J. Hering, “A virtual flood information: open data for sustainable water management,” 2014. .
- [67] “GEOSS - The Global Earth Observation System of Systems.” .
- [68] “GEO Data Sharing Principles Implementation.” .
- [69] “Earth System Science Data - The Data Publishing Journal,” 2014. .
- [70] “Policy Recommendations for Open Access to Research Data in Europe.” .
- [71] “Linked-data UI’s: a response to David Karger,” 2013. .
- [72] S. Peroni, D. Shotton, and F. Vitali, “Making Ontology Documentation with {LODE},” in *Proceedings of the {I-SEMANTICS} 2012 Posters {&} Demonstrations Track, Graz, Austria, September 5-7, 2012*, 2012, vol. 932, pp. 63–67.
- [73] S. Lohmann, S. Negru, F. Haag, and T. Ertl, “VOWL 2: User-Oriented Visualization of Ontologies,” in *Knowledge Engineering and Knowledge Management - 19th International Conference, {EKAW} 2014, Link{ö}ping, Sweden, November 24-28, 2014. Proceedings*, 2014, vol. 8876, pp. 266–281.
- [74] C. Baldassarre, E. Daga, A. Gangemi, A. M. Gliozzo, A. Salvati, and G. Troiani, “Semantic Scout: Making Sense of Organizational Knowledge.,” in *EKAW*, 2010, pp. 272–286.
- [75] “The WISDOM Project.” .
- [76] “sMAP: the Simple Measurement and Actuation Profile.” .
- [77] “sMAP 2.0 Documentation.” .
- [78] “Twisted.” .
- [79] “Readingdb time series database.” .
- [80] “Pachube.” .
- [81] B. Kämpgen, D. Riepl, and J. Klinger, “SMART Research using Linked Data - Sharing Research Data for Integrated Water Resources Management in the Lower Jordan Valley.,” in *SePublica*, 2014.
- [82] R. Cyganiak and D. Reynolds, “The RDF Data Cube Vocabulary.” .
- [83] “The SMART II Project.” .
- [84] “Dropedia, the collaborative knowledge management platform about Integrated Water Resources Management in the Lower Jordan Rift Valley.” .