The Copernicus App Lab project: Easy Access to Copernicus Data

Konstantina Bereta
National and Kapodistrian University of Athens
Greece
konstantina.bereta@di.uoa.gr

Hervé Caumont
Terradue Srl
Italy
herve.caumont@terradue.com

Ulrike Daniels
AZO Anwendungszentrum GmbH
Germany
Ulrike.Daniels@azo-space.com

Daems Dirk
VITO
Belgium
dirk.daems@vito.be

Erwin Goor*
European Commission
Belgium
Erwin.GOOR@ec.europa.eu

Manolis Koubarakis
National and Kapodistrian University of Athens
Greece
koubarak@di.uoa.gr

Despina-Athanasia Pantazi
National and Kapodistrian University of Athens
Greece
dpantazi@di.uoa.gr

George Stamoulis
National and Kapodistrian University of Athens
Greece
gstam@di.uoa.gr

Sam Ubels
RAMANI B.V.
The Netherlands
sam.ubels@ujuizi.com

Valentijn Venus
RAMANI B.V.
The Netherlands
valentijn.venus@ujuizi.com

Firman Wahyudi
RAMANI B.V.
The Netherlands
firman.wahyudi@ujuizi.com

ABSTRACT

Copernicus is the European programme for monitoring the Earth. It consists of a set of complex systems that collect data from satellites and in-situ sensors, process this data, and provide users with reliable and up-to-date information on a range of environmental and security issues. Information extracted from Copernicus data is made available to users through Copernicus services addressing six thematic areas: land, marine, atmosphere, climate, emergency and security. The data and information processed and disseminated puts Copernicus at the forefront of the big data paradigm and gives rise to all relevant challenges: volume, velocity, variety, veracity and value. In this paper we discuss the challenges of big Copernicus data and how the Copernicus programme is attempting to deal with them. We also present lessons learned from our project Copernicus App Lab, which takes Copernicus services information and makes it available on the Web using semantic technologies to aid its take up by mobile developers. We also discuss open problems for information retrieval, database and knowledge management researchers in the context of Copernicus.

1 INTRODUCTION

Earth observation (EO) is the gathering of data about our planet's physical, chemical and biological systems via satellite remote sensing technologies supplemented by Earth surveying techniques. The Landsat program of the US was the first international program that made large amounts of EO data open and freely available. *Copernicus*, the European programme for monitoring the Earth, is currently the world's biggest EO programme. It consists of a set of complex systems that collect data from satellites and in-situ sensors, process this data and provide users with reliable and up-to-date information on a range of environmental and security issues. The data of the Copernicus programme is provided by a group of missions created by ESA, which is called *Sentinels*, and the *contributing missions*, which are operated by national, European or international organizations. Copernicus data is made available under a free, full and open data policy. Information extracted from this data is also made freely available to users through the *Copernicus services* which address six thematic areas: land, marine, atmosphere, climate, emergency and security.

The Copernicus programme offers myriad forms of data that enable citizens, businesses, public authorities, policy makers, scientists, and entrepreneurs to gain insights into our planet on a free, open, and comprehensive basis. By making the vast majority of its data, analyses, forecasts, and maps freely available and accessible, Copernicus contributes to the development of innovative applications and services that seek to make our world safer, healthier, and economically stronger. However, the potential (in both societal and economic terms) of these huge amounts of data can only be fully exploited if using them is made as simple as possible. Therefore, the straightforward data access every downstream service developer requires must also be combined with in-depth knowledge of EO data processing. The Copernicus App Lab[1] aims to address these specific challenges by bridging the digital divide between the established, science-driven EO community and the young, innovative, entrepreneurial world of mobile development.

*Work performed while at VITO

[1] http://www.app-lab.eu/

Copernicus App Lab is a two year project (November 2016 to October 2018) funded by the European Commission under the H2020 programme. The consortium consists of the company AZO[2] (project coordinator), the National and Kapodistrian University of Athens[3], the companies Terradue[4] and RAMANI[5] and the Flemish research institute VITO[6]. The main objective of Copernicus App Lab is to make Earth observation data produced by the Copernicus programme available on the Web as *linked data* to aid its use by users that might not be Earth observation experts.

Copernicus App Lab targets the *volume* and *variety* challenges of Copernicus data, and it follows the path of previous research projects TELEIOS[7], LEO[8] and MELODIES[9]. Under the lead of the National and Kapodistrian University of Athens, these three projects pioneered the use of linked geospatial data in the EO domain, and demonstrated the potential of linked data and semantic web technologies in the Copernicus setting by developing prototype environmental and business applications (e.g., wild-fire monitoring and burn scar mapping [18, 20], precision farming [9], maritime security [8] etc.).

Copernicus App Lab goes beyond these projects in the following important ways:

- It develops a software architecture that enables on demand access to Copernicus data using the well-known OPeNDAP framework and the geospatial ontology-based data access system Ontop-spatial [5]. Now users and application developers do not need to worry about having to download data or having to learn the details of sophisticated data formats for EO data.
- It brings computing resources close to the data by making the Copernicus App lab tools available as Docker images that are deployed in the Terradue cloud platform as cloud services. The platform allows application developers to access Copernicus data and carry out massively parallel processing without the need to download the data in their own servers and carry out the processing locally.
- It enables search engines like Google to treat datasets produced by Copernicus as "entities" in their own right and store knowledge about them in their internal knowledge graph. In this way, search engines will be able to answer sophisticated users questions involving datasets such as the following: "Is there a land cover dataset produced by the European Environmental Agency covering the area of Torino, Italy?"

A demo paper describing the application scenario presented in this paper won the "best demo award" prize at CIKM 2018 [3]. Shorter presentations of the Copernicus App Lab project also appears in [2]. Compared with the present paper, these papers emphasize only the semantic technologies developed and only the big data dimensions respectively.

The above innovations of Copernicus App Lab are discussed in detail in the rest of the paper, which is organized as follows. Section 2 presents the related work. Section 3 presents the conceptual architecture of the Copernicus integrated ground segment and the software architecture of Copernicus App Lab. Section 4 presents a simple case study which demonstrates the technologies of Copernicus App Lab. Section 5 discusses lessons learned by the use of these technologies and discusses open problems that need to be tackled by future research. Finally, Section 6 concludes the paper.

[2]http://www.anwendungszentrum.de/
[3]http://kr.di.uoa.gr/
[4]https://www.terradue.com/
[5]https://ramani.ujuizi.com/
[6]https://remotesensing.vito.be/
[7]http://www.earthobservatory.eu/
[8]http://www.linkedeodata.eu/
[9]https://www.melodiesproject.eu/

2 RELATED WORK

Open EO data that are currently made available by the Copernicus and Landsat programs are not following the linked data paradigm. Therefore, from the perspective of a user, the EO data and other kinds of geospatial data necessary to satisfy his or her information need can only be found in different data silos, where each silo may contain only part of the needed data. Opening up these silos by publishing their contents as RDF and interlinking them with semantic connections will allow the development of data analytics applications with great environmental and financial value.

Previous projects TELEIOS, LEO and Melodies funded by FP7 ICT, have demonstrated the use of linked data in Earth Observation. The European project TELEIOS was the first project internationally that has introduced the linked data paradigm to the EO domain, and developed prototype applications that are based on transforming EO products into RDF, and combining them with linked geospatial data. TELEIOS concentrated on developing data models, query languages, scalable query evaluation techniques, and efficient data-management systems that can be used to prototype the applications of linked EO data [18]. The European project LEO was to go beyond TELEIOS by designing and implementing software supporting the complete life cycle of linked open EO data and its combination with linked geospatial data and by developing a precision farming application that heavily utilizes such data [9]. The MELODIES project developed new data-intensive environmental services based on data from Earth Observation satellites, government databases, national and European agencies and more [7]. We focused on the capabilities and benefits of the project's "technical platform", which applied cloud computing and Linked Data technologies to enable the development of services, providing flexibility and scalability.

One of the most important objectives achieved by the aforementioned projects, was capturing the life cycle of open EO data and the associated entities, roles and processes of public bodies that make this data available, and bring the linked data paradigm to EO data centers by re-engineering a complete data science pipeline for EO data [16, 19].

3 THE COPERNICUS APP LAB ARCHITECTURE

Figure 1 presents the conceptual architecture of the *Copernicus integrated ground segment* and the Copernicus App Lab software architecture. A *ground segment* is the hardware and software infrastructure where raw data, often from multiple satellite missions, is ingested, processed, cataloged, and archived. The processing results in the creation of various *standard products* (level 1, 2, and so forth in EO jargon; raw data is level 0) together with extensive metadata describing them.

In the lower part of the figure, the Copernicus *data sources* are shown. These are Sentinel data from ESA, Sentinel data from the European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT), satellite data from contributing missions

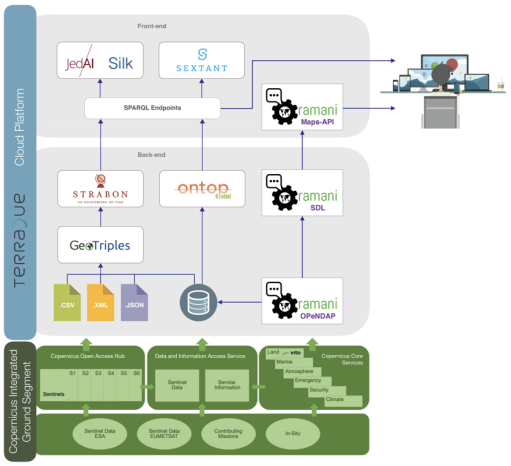

Figure 1: The Copernicus integrated ground segment and the Copernicus App Lab software architecture

(e.g., the RADARSAT-2 mission of Canada or the PROBA-V mission of Belgium) and in-situ data (e.g., data from sensors measuring major air pollutants). The next layer makes Copernicus data and information available to interested parties in three ways: via the Copernicus Open Access Hub, via the Copernicus Core Services and via the Data and Information Access Service (most often known by its acronym DIAS).

The Copernicus Open Access Hub[10] is currently the primary means of accessing Sentinel data. It offers a simple graphical interface that enables users to specify the extent of the geographical area one is interested in (by drawing a bounding box or a polygon). The user may also complete a search interface with information regarding the sensing period, the satellite, the platform, the sensor mode, the polarization etc. If a relevant product is found, the product can be downloaded to the user's computer.

The six core Copernicus services (land, marine, atmosphere, climate, emergency and security) are administered by European entrusted entities (e.g., the Copernicus App Lab partner VITO administers the global land service), whose job is to process Copernicus data and produce higher-level products (*information* in the Copernicus jargon) that are of importance in the corresponding thematic area (e.g., leaf-area index data in the case of the global land service).

The DIAS is not yet fully developed. In December 2017, the European Commission has awarded four contracts to industrial consortia for the development of four cloud-based DIAS platforms. A fifth DIAS is developed by EUMETSAT in collaboration with the French company Mercator Ocean and the European Center for Medium-Range Weather Forecasts (ECWMF). The five DIAS will bring computing resources close to the data and enable an even greater commercial exploitation of Copernicus data. The first versions of the five DIAS are now open to demo-users.

As we have already mentioned in the introduction, the goal of Copernicus App Lab is to make earth observation data produced by the Copernicus programme available as linked data to aid its use by developers that might not be experts in Earth Observation. The software architecture presented in the top two layers of Figure 1 has been designed for achieving this goal. Since none of the DIAS platforms was available when Copernicus App Lab started, all the software components of the project run in the Terradue cloud platform[11]. Terradue Cloud Platform is built as a Hybrid Cloud platform. The primary purpose of the Hybrid Cloud Platform is to facilitate the management of elastic compute resources, with low cost scale-out capabilities. It relies on the concept of an application integration environment (PaaS, or Platform-as-a-Service) and a production environment. Terradue Cloud Platform builds on three major outcomes of the recent developments in Computer Science and Web technology - Cloud Computing, Open Data repositories, and Web Services interoperability. The platform allows cloud orchestration, storage virtualisation, and virtual machine provisioning, as well as application burst-loading and scaling on third-party cloud infrastructures. Within the Terradue cloud platform, the developer cloud sandbox service provides a platform-as-a-service (PaaS) environment to prepare data and processors. It has been designed with the goal to automate the deployment of the resulting EO applications to any cloud computing facility that can offer storage and computing resources (e.g., Amazon Web services). In this manner, the AppLab Cloud Architecture provides the infrastructure (cloud environment) to bring together all the elements of the Copernicus App Lab, and ensure operations.

In Copernicus App Lab, access to Copernicus data and information can be achieved in two ways: (i) by downloading the data via the Copernicus Open Access Hub or the Web sites of individual Copernicus services, and (ii) via the popular OPeNDAP framework[12] for accessing scientific data. In the first case (workflow on the left part of the two top layers of Figure 1), the downloaded data should then transformed into RDF using the tool GeoTriples [23] or scripts written especially for this task. GeoTriples enables the transformation of geospatial data stored in raw files (shapefiles, CSV, KML, XML, GML and GeoJSON) and spatially-enabled RDBMS (PostGIS and MonetDB) into RDF graphs using well-known geospatial vocabularies e.g., the vocabulary of the Open Geospatial Consortium (OGC) standard GeoSPARQL [27]. The performance of GeoTriples has been studied experimentally in [22] using large publicly available geospatial datasets. It has been shown that GeoTriples is very efficient especially when its mapping processor is implemented using Apache Hadoop.

After Copernicus data has been transformed into RDF, it can be stored in the spatiotemporal RDF store Strabon [6, 21]. Strabon can store and query linked geospatial data that changes over time. It has been shown to be the most efficient spatiotemporal RDF store available today using the benchmark Geographica in [6, 15].

Copernicus data stored in Strabon may also be interlinked with other relevant data (e.g., a dataset that gives the land cover of certain areas might be interlinked with OpenStreetMap data for the same areas). To do this in Copernicus App Lab, we use the interlinking tools JedAI and Silk. JedAI is a toolkit for entity resolution and its multi-core version has been shown to be scalable to very large datasets [25]. Silk is a well-known framework for interlinking RDF datasets which we have extended to deal with geospatial and temporal relations [28].

The above way of accessing and using Copernicus data as linked data, has been introduced in previous projects TELEIOS, LEO and MELODIES discussed in the introduction, and it is not the focus of this paper. The *novel way of accessing Copernicus data and information* in Copernicus App Lab is captured by the

[10] https://scihub.copernicus.eu/

[11] https://www.terradue.com/portal/

[12] https://www.opendap.org/

workflow on the right part of the two top layers of Figure 1, and it is based on the popular *OPeNDAP framework* for accessing scientific data. OPeNDAP provides a powerful data translation facility so that users do not need to know the detailed formats of data stored in servers, and can simply use a client that supports a model they are comfortable with. The *streaming data library (SDL)* implemented by RAMANI communicates with the OPeNDAP server and receives Copernicus services data as *streams*. In the rest of this section, we describe how we can access Copernicus data on-the-fly using this workflow.

3.1 Enhance Cataloguing of Copernicus data streams

Copernicus Service Providers (CSP) publish their data holdings in a variety of data formats and access protocols (ftp, http, dap), with various metadata co-existing, and data and metadata either separate or combined in one container. In order to enrich the metadata coupled with the CSP's offerings, we set a minimum metadata standard which should be followed by interested parties in order to streamline the classification, mapping and RDF linkage. We present a mediation approach that facilitates multiple Metadata Standards to co-exist but are semantically harmonized through SPARQL Query. A command-line tool was build and published, entitled "DRS-validator", that validates a CSP's datasets exposed through the OPeNDAP interface by checking for compliance with the Data Reference Syntax (DRS) metadata. Given the proliferation of various metadata standards, a tool was developed that can translate between metadata conventions. In order to harvest the metadata, a Content Management System (CMS) was developed and published as a service allowing the CSP's to manage the metadata of their datasets, which allows them to mutate as and when they choose to expose them through the DAP. Completeness of metadata can be checked globally at SDL level or at an individual dataset level. Since we also use the netCDF variable attributes and global attributes to perform machine-to-machine communication of metadata, the publishing and then harvesting of metadata from CSPs is recurrent by design. For communicating metadata, we use the NetCDF Markup Language (NcML) interface service. This extends a dataset's OPeNDAP Dataset Attribute Structure (DAS) and Dataset Descriptor Structure (DDS) into a single XML-formatted document. The DDS describes the dataset's structure and the relationships between its variables, and the DAS provides information about the variables themselves. The returned document may include information about both the data server itself (such as server functions implemented), and the metadata and dataset referenced in the URL. To ensure that metadata contributes to the discoverability of a datasets, a tool was implemented that provides recommendations for metadata attributes that can be added to datasets exposed through the DAP to facilitate discovery of those using standard metadata searches.

Depending on the requirements imposed upon us by the other open-data, additional analytical functionality is created to ensure the geospatial data streams are commensurate with these requirements. We added a software layer to the SDL, entitled RAMANI Cloud Analytics, allowing on-the-fly spatial and temporal aggregations such that downstream services may request for derived variables to be returned, such as a long-term (moving) average (summer-time) or spatial central tendency (city-average), ensuring that the return is fully commensurate with the intrinsic requirements of the other data to be linked. The technology being used in this stage consists of background IP (BIPR) of RAMANI B.V., with the development of an additional layer for supporting Open Linked Data and Web Coverage Service (WCS). We designed a backend solution capable of providing analytics on data via linked relations. Basic analytical modelling processes can be represented straightforwardly by starting with data and do calculations/analysis that result in more data, which we eventually synthesize into a pithy result, like part of an App or a presentation. Also, the analysis can easily be rerun, if, for example, the data is extended in time, otherwise modified, or is replaced by a different data source providing similar variables based on semantically provided heuristics (e.g. based on "hasName" or "hasUnit"). To allow fast processing we developed a set of containers of the server-side analytics package. Then we used Kubernetes[13] for managing the containerized applications across multiple hosts, that provides the mechanisms for deployment, maintenance, and scaling of the RAMANI Cloud Analytics backend services. Kubernetes builds upon a decade and a half of experience at Google running production workloads at scale using a system called Borg, combined with best-of-breed ideas and practices from the community.

The SDL is also extended with Semantic Web standards providing a framework for explicitly describing the data models implicit in EO and other GRIDded data streams to enhance downstream display and manipulation of data. This provides a framework where multiple metadata standards can be described. Most importantly, these data models and metadata standards can be interrelated, a key step in creating interoperability, and an important step in being able to map various metadata formats in compliance with other standards, e.g. as put forward by the INPIRE Spatial Data Services Working Group. First we developed an Abstract Model capable of providing the basis of the semantics for the linked RDFS. Based on a proof-of-concept use case, an example of a RDF/XML expression for a remote OPeNDAP dataset[14] sourced from the Copernicus Land Monitoring programme was created. This dataset is published using a local deployment of the DAP at VITO. We then implemented a vetted RDF crawler that handles non-standard metadata and supports reasoners, query languages, parsers and serializers. The query languages can create new triples based on query matches (CONSTRUCT) and reasoners create virtual triples based on the stated interrelationships, so we have a framework for creating crosswalks between metadata standards, as well as creating code that is independent of the metadata standards. Finally, in case metadata at the source cannot be made compliant with ACDD, the CMS will allow for post-hoc augmentation using NcML blending metadata provided by the source and those required as-per the DRS validator.

In the current version of the Copernicus App Lab software, VITO -as prime contractor of the Copernicus Global Land Service- has engaged with RAMANI and now provides access to the Copernicus Global Land Service using a remote data access protocol facilitating discovery, viewing, and access of their unique Land Monitoring data-products. OPeNDAP and SDL are installed and configured by VITO on a virtual machine running on the VITO hosted PROBA-V mission exploitation platform[15], which has direct access to the data archives of the Copernicus global land service. The original data sources (i.e. Leaf Area Index, from PROBA-V) have been added to the Streaming Data Library (SDL) so their temporal and spatial characteristics are exposed in a queryable manner. Three Copernicus datasets as well as one

[13] http://kubernetes.io
[14] http://land.copernicus.eu/global/products/lai
[15] https://proba-v-mep.esa.int

Proba-V dataset were configured in the initial setup. These are BioPar BA300 (Burnt Area), LAI (Leaf Area Index) and NDVI (Normalised Difference Vegetation Index). The S5 (five-daily composite) TOC (Top of Canopy, i.e. after atmospheric correction) NDVI 100M was supposed to be implemented as a Proba-V dataset. Each dataset also contains a netCDF NCML aggregation, which is automatically updated when new data (a new date) becomes available. Three different services are exposed for each dataset: the OPeNDAP service, the NetcdfSubset service and the NCML service. The installation of OPeNDAP was done using Docker and access to the Copernicus global land and PROBA-V datasets via OPeNDAP is realised by mounting the necessary disks on the virtual machine.

3.2 Querying Copernicus data on-the-fly using GeoSPARQL

The geospatial ontology-based data access (OBDA) system Ontop-spatial [4] is used to make Copernicus data available via OPeNDAP as linked geospatial data, without the need for downloading files and transforming them into RDF. Ontop-spatial is the geospatial extension of the OBDA system Ontop[16] [10]. OBDA systems are able to connect to existing, relational data sources and create virtual semantic graphs on top of them using ontologies and mappings. An ontology provides a conceptual view of the data. Mappings encode how relational data can be mapped into the terms described in the ontologies. The mapping language R2RML is a W3C standard and is commonly used to encode mappings, but a lot of OBDA/RDB2RDF systems also offer a native mapping language. Since Ontop-spatial follows the OBDA paradigm in the geospatial domain, it can be used to create virtual semantic RDF graphs on top of geospatial relational data sources using ontologies and mappings.

Then, the Open Geospatial Consortium standard GeoSPARQL can be used to pose queries to the data using the ontology. As documented in [4], Ontop-spatial also achieves significantly better performance than state-of-the-art RDF stores. An extension of this work described in [5] shows how the system was extended to support raster data sources as well (support for raster data sources has not been foreseen in the GeoSPARQL standard) and it also describes how, using the OBDA approach, one can query both vector and raster data sources combined in a purely transparent way, without the need to extend the GeoSPARQL query language further.

In the context of the work described in this paper, we extend the OBDA paradigm even more, by enabling an OBDA system not only to connect to a non-relational data source, but also to query data sources that are available remotely, without accessing or storing the data locally (e.g., import the datasets into database); the data can be available through a REST API, for example, and can be accessed by the system only after a GeoSPARQL query is fired. Ontop-spatial has been extended with an adapter that enables it to retrieve data from an OPeNDAP server, create a table view on-the-fly, populate it with this data and create virtual semantic geospatial graphs on top of them. In order to use OPeNDAP as a new kind of data source, Ontop-spatial utilizes the system MadIS[17] as a back-end. MadIS is an extensible relational database system built on top of the APSW SQLite wrapper [11]. It provides a Python interface so that users can easily implement user-defined functions (UDFs) as rows, aggregate functions, or virtual tables. We used MadIS to create a new UDF, named `Opendap`, that is able to create and populate a virtual table on-the-fly with data retrieved from an OPeNDAP server. In this way, Ontop-spatial enables users to pose GeoSPARQL queries on top of OPeNDAP data sources without materializing any triples or tables. We stress that the relational view that is created is not materialized. The intermediate SQL layer facilitates the data manipulation process, in the sense that we can manipulate the data before they get RDF-ized. In this way we can perform "data cleaning" without (i) changing the data that arrive from the server (ii) changing any intermediate code, such as the opendap function and (ii) without requiring any extra pre-processing steps. In order to be able to integrate MadIS as back-end system of Ontop-spatial, apart from implementing the OPeNDAP virtual table function, we also had to extend its jdbc connector and make several modifications in the core of Ontop code so that it allows connections to non-relational data. The design choice behind our approach to implement the OPeNDAP adapter as a virtual table is that it improves the extensibility of the system, as more adapters could be added in the same way by implementing more UDFs in the MadIS system. The integration of the MadIS system to Ontop-spatial is a gateway to support any API in the future that could serve as a new data source to the system.

To improve performance, the OPeNDAP adapter also implements a caching mechanism that stores results of an OPeNDAP call in a cache for a time window w, so that if another, identical OPeNDAP call needs to be performed within this time window, the cached results can be used directly. The length of the time window w is configured by the user in the mappings and it is optional. In the context of the application scenario described in Section 4, we provide more details about the implementation and use of this new version of Ontop-spatial.

3.3 Visualization

Data can be visualized using the tools Sextant [24] or Maps-API.[18] Sextant is a web-based and mobile ready application for exploring, interacting and visualizing time-evolving linked geospatial data. What we wanted to achieve is develop an application that is flexible, portable and interoperable with other GIS tools. The core feature of Sextant is the ability to create thematic maps by combining geospatial and temporal information that exists in a number of heterogeneous data sources ranging from standard SPARQL endpoints, to SPARQL endpoints following the standard GeoSPARQL defined by the Open Geospatial Consortium (OGC), or well-adopted geospatial file formats, like KML, GML and GeoTIFF. In this manner we provide functionality to domain experts from different fields in creating thematic maps, which emphasize spatial variation of one or a small number of geographic distributions. Each thematic map is represented using a map ontology that assists on modelling these maps in RDF and allow for easy sharing, editing and search mechanisms over existing maps.

The Maps-API is similar to Sextant in terms of visualization functionality, but it takes its data from SDL and it cannot deal with linked geospatial data sources accessed by SPARQL or GeoSPARQL. Once data has been discovered, it can be consumed in the VISual Maps-API using any of the following data request-methods: getMetadata, getDerivedData, getMap, getAnimation, getTransect, getPoint, getArea, getVerticalProfile, getSpectralProfile (in case of multi-spectral EO-data), getMapSwipe, and getTimeseriesProfile. This is mainly intended for App Developers who wish to

[16] http://ontop.inf.unibz.it/
[17] https://github.com/madgik/madis
[18] https://ramani.ujuizi.com/maps/index.html

integrate and consume the Copernicus services' products in their favourite mobile platform(s) using straightforward visualization, e.g. as layers on a map-view or as independent graphics (w/o associated geometries on the map).

A more detailed discussion of the linked data tools introduced above and their use in the life cycle of linked Earth observation data is given in the survey papers [14, 17]. All tools are open source and they are available on the following Web page: http://kr.di.uoa.gr/#systems

4 A COPERNICUS APP LAB CASE STUDY

We will now present a simple case study which demonstrates the functionality of the Copernicus App Lab software presented in the previous section. The case study is the same as the scenario presented in the demo paper [3]. However, the presentation here is much more detailed and concentrates on the technical challenges and contributions of the case study.

The case study involves studying the "greenness" of Paris. This can be done by relating "greenness" features of Paris using geospatial data sources such as OpenStreetMap (e.g., for features like parks and forests) and relevant Copernicus datasets. The most important source of such data in Copernicus is the *land monitoring service*.[19] The data provided by this service belongs to the following categories:

- *Global*: The Copernicus Global Land Service (CGLS) is a component of the Land Monitoring Core Service (LMCS) of Copernicus, the European flagship programme on Earth Observation. The Global Land Service systematically produces a series of qualified bio-geophysical products on the status and evolution of the land surface, at global scale and at mid to low spatial resolution, complemented by the constitution of long term time series. The products are used to monitor the vegetation, the water cycle, the energy budget and the terrestrial cryosphere. These datasets are provided by the Copernicus App Lab partner VITO via its satellite PROBA-V. They include a series of bio-geophysical products on the status and evolution of the Earth's land surface at global scale at mid and low spatial resolution.
- *Pan-European*: The pan-European component is coordinated by the European Environment Agency (EEA) and produces satellite image mosaics, land cover / land use (LC/LU) information in the CORINE Land Cover data, and the High Resolution Layers. The CORINE Land Cover is provided for 1990, 2000, 2006 and 2012. This vector-based dataset includes 44 land cover and land use classes. The time-series also includes a land-change layer, highlighting changes in land cover and land-use. The high-resolution layers (HRL) are raster-based datasets which provides information about different land cover characteristics and is complementary to land-cover mapping (e.g. CORINE) datasets. Five HRLs describe some of the main land cover characteristics: impervious (sealed) surfaces (e.g. roads and built up areas), forest areas, (semi-) natural grasslands, wetlands, and permanent water bodies. The High-Resolution Image Mosaic is a seamless pan-European ortho-rectified raster mosaic based on satellite imagery covering 39 countries.
- *Local*: The local component is coordinated by the European Environment Agency and aims to provide specific and more detailed information that is complementary to the information obtained through the Pan-European component. The local component focuses on different hotspots, i.e. areas that are prone to specific environmental challenges and problems. It will be based on very high resolution imagery (2,5 x 2,5 m pixels) in combination with other available datasets (high and medium resolution images) over the pan-European area. The three local components are Urban Atlas, Riparian Zones and Natura 2000.
- *Reference data*: Copernicus land services need both satellite images and in-situ data in order to create reliable products and services. Satellite imagery forms the input for the creation of many information products and services, such as land cover maps or high resolution layers on land cover characteristics. Having all the satellite imagery available to cover 39 countries of EEA (EEA39), the individual image scenes have been processed into a seamless pan-European ortho-rectified mosaics. The Copernicus Land Monitoring Service also provides access to Sentinel-2 Global Mosaic service. A lot of in-situ data is managed and made accessible at national level. However, due to issues such as data access and use restrictions, data quality and availability across EEA39 countries, Copernicus services and particularly Copernicus Land Monitoring Service also relies on pan-European in-situ datasets created and/or coordinated at European level. These datasets are needed for the verification and validation of satellite data in the land monitoring service portfolio.

For our case study, the most relevant datasets from the land monitoring service of Copernicus are the leaf-area index dataset (global), the CORINE land cover dataset (pan-European) and the Urban Atlas dataset (local).

Leaf area index (LAI) is a dimensionless quantity that characterizes plant canopies and it is defined as the one-sided green leaf area per unit ground surface area in broadleaf canopies[20]. LAI may range from 0 (bare ground) to 10 (dense coniferous forests). LAI information from the global land service of Copernicus is made available as a NetCDF file giving LAI values for points expressed by their lat/long co-ordinates.

The *CORINE land cover dataset* in its most recent version (2012) covers 39 European countries[21]. Land cover is characterized using a 3-level hierarchy of classes (e.g., olive groves or vineyards) with 44 classes in total at the 3rd level. The minimum mapping unit is 25 hectares for areal phenomena and 100 meters for linear phenomena. It is made available in raster (GeoTIFF) and vector (ESRI/SQLite geodatabase) formats.

The *Urban Atlas dataset* in its most recent version (2012) provides land use and land cover data for European urban areas with more than 100.000 inhabitants[22]. It covers 800 urban areas in 28 European Union countries, the 4 European Union Free Trade Association countries (Switzerland, Iceland, Norway and Lichtenstein), Turkey and the West Balkans. Land cover/land use is characterized by 17 urban classes (e.g., discontinuous very low density urban fabric) with minimum mapping unit 0.25 hectares, and 10 rural classes (e.g., orchards) with minimum mapping unit 1 hectare. It is made available in vector format as ESRI shapefiles.

In addition to the above datasets, our case study utilizes data from OpenStreetMap and the global administrative divisions dataset GADM. OpenStreetMap is an open and free map of the

[19] https://land.copernicus.eu/

[20] https://en.wikipedia.org/wiki/Leaf_area_index

[21] https://land.copernicus.eu/pan-european/corine-land-cover/view

[22] https://land.copernicus.eu/local/urban-atlas/view

whole world constructed by volunteers. It is available in vector format as shapefiles from the German company Geofabrik[23]. For our case study, information about parks in Paris has been taken from this dataset.

GADM is an open and free dataset giving us the geometries of administrative divisions of various countries[24]. It is available in vector format as a shapefile, a geopackage (for SQLlite3), a format for use with the programming language R, and KMZ (compressed KML).

The first task of any case study using the Copernicus App Lab software is to develop INSPIRE-compliant ontologies for the selected Copernicus data. The *INSPIRE directive* aims to create an interoperable spatial data infrastructure for the European Union, to enable the sharing of spatial information among public sector organizations and better facilitate public access to spatial information across Europe[25]. *INSPIRE-compliant ontologies* are ontologies which conform to the INSPIRE requirements and recommendations. The INSPIRE directives provide a set of data specifications for a wide variety of themes. Our purpose is to categorize our datasets into INSPIRE themes, construct an ontology that follows the respective data specification and then extend this generic ontology to create a specialized version in order to model our datasets. Our initial approach was to reuse existing inspire-compliant ontologies , such as the ones described in [26], but since these efforts are not as close to the INSPIRE specifications as we would like to, we decided to create our own INSPIRE-compliant versions, following the data specifications as closely as possible. Our aim is to reuse these ontologies for other datasets that belong to the same INSPIRE themes and also publish them so that others can reuse these ontologies for their geospatial datasets as well.

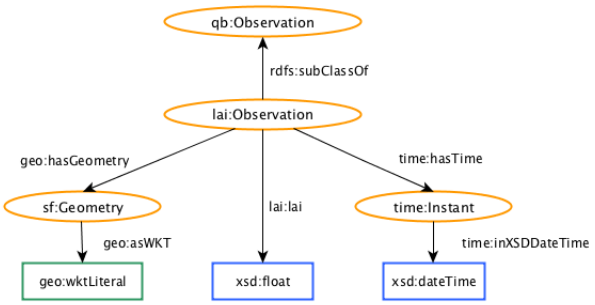

Figure 2: The LAI ontology

A simple ontology for the LAI dataset is shown in Figure 2. We have re-used classes and properties from the Data Cube ontology[26] (namespace `qb`) specializing them when appropriate. We also used classes and properties from the GeoSPARQL ontology [27] (namespaces `sf` and `geo`), from the Time Ontology[27] (namespace `time`), and datatypes from XML-Schema. The class and properties introduced by us use the prefix `lai`.

Once the LAI ontology is created, a user following the workflow depicted on the left, in the Copernicus App Lab software architecture of Figure 1, can use it to transform into RDF the most recent LAI dataset that is made available by the Copernicus global land service[28]. Since GeoTriples does not support NetCDF files as input, the translation was done by writing a custom Python script.

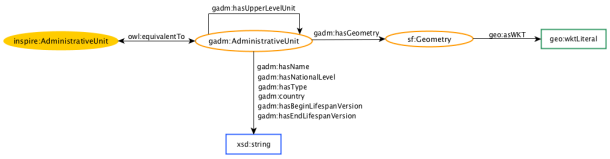

Figure 3: The GADM ontology

Figure 3 shows the ontology that we created for the GADM dataset by extending the GeoSPARQL ontology [27] (namespaces `sf` and `geo`). For the class and properties that we introduced we use the prefix `gadm`[29].The GADM ontology can be used so that a GADM dataset[30] can be either converted into RDF or queried on-the-fly.

A similar process can be followed for datasets CORINE land cover, Urban Atlas, and OpenStreetMap. The ontology which we constructed for the CORINE land cover dataset is not shown here due to space considerations but it is available online [31]. Among other entities (i.e., a class hierarchy corresponding to CORINE land cover categories), it includes the following elements:

- `clc:CorineArea`. This class is a subclass of the class `inspire:LandCoverUnit` of the INSPIRE theme for land cover.
- `clc:hasCorineValue`. This property associates a `clc:CorineArea` with its land cover, as characterised by CORINE.
- `clc:CorineValue`. This class is the range of the property `clc:hasCorineValue` and it is superclass of all the land cover classes of the CORINE hierarchy e.g., `clc:Forests`.

The CORINE land cover ontology can be used to model CORINE land cover data, either materialised as RDF dumps, or virtual RDF graphs created by an OBDA system.

Similarly, we have defined ontologies for Urban Atlas[32], and OpenStreetMap[33]. In the past OpenStreetMap data have been made available in RDF by project LinkedGeoData[34] [29] in the context of which a SPARQL endpoint for OpenStreetMap data was also created. However, the data in this endpoint is not up-to-date (the current version is from 2015), and also this endpoint does not support GeoSPARQL queries. Therefore, we constructed a new ontology for OpenStreetMap by following closely the description of OpenStreetMap data provided by Geofabrik[35] and made it available also in OWL[36]. Using this ontology, we have transformed OpenStreetMap data in shapefiles format into RDF using the tool GeoTriples.

Once all the above datasets are available in RDF, they can be stored in Strabon enabling users to pose interesting, rich queries against the combined dataset. For example, assuming appropriate PREFIX definitions, the GeoSPARQL query shown in Listing 1

[23] http://download.geofabrik.de/
[24] https://gadm.org/
[25] https://inspire.ec.europa.eu/
[26] https://www.w3.org/TR/vocab-data-cube/
[27] https://www.w3.org/TR/owl-time/
[28] https://land.copernicus.eu/global/products/lai
[29] The corresponding namespace is: http://www.app-lab.eu/gadm/
[30] https://gadm.org/data.html
[31] http://pyravlos-vm5.di.uoa.gr/corineLandCover.svg
[32] http://pyravlos-vm5.di.uoa.gr/urbanOntology.svg
[33] http://sites.pyravlos.di.uoa.gr/dragonOSM.svg
[34] http://linkedgeodata.org/About
[35] http://download.geofabrik.de/osm-data-in-gis-formats-free.pdf
[36] http://pyravlos-vm5.di.uoa.gr/osm.owl

retrieves the LAI values of the area occupied by the Bois de Boulogne park in Paris.

Listing 1: LAI in Bois de Boulogne

```
SELECT DISTINCT ?geoA ?geoB ?lai WHERE
{  ?areaA osm:poiType osm:park .
   ?areaA geo:hasGeometry ?geomA .
   ?geomA geo:asWKT ?geoA .
   ?areaA osm:hasName
   "Bois de Boulogne"^^xsd:string> .
   ?areaB lai:lai ?lai .
   ?areaB geo:hasGeometry ?geomB .
   ?geomB geo:asWKT ?geoB .
    FILTER(geof:sfIntersects(?geoA, ?geoB))
}
```

Similarly, in Figure 4, we have used Sextant to build a temporal map that shows the "greenness" of Paris, using the datasets LAI, GADM, CORINE land cover, Urban Atlas and OpenStreetMap. We show how the LAI values (small circles) change over time in each administrative area of Paris (administrative areas are delineated by magenta lines) and correlate these readings with the land cover of each area (taken from the CORINE land cover dataset or Urban Atlas). This allows us to explain the differences in LAI values over different areas. For example, Paris areas belonging to the CORINE land cover class `clc:greenUrbanAreas` overlap with parks in OpenStreetMap and show higher LAI values over time than industrial areas. Paris enthusiasts are invited to locate the Bois de Boulogne park in the figure.

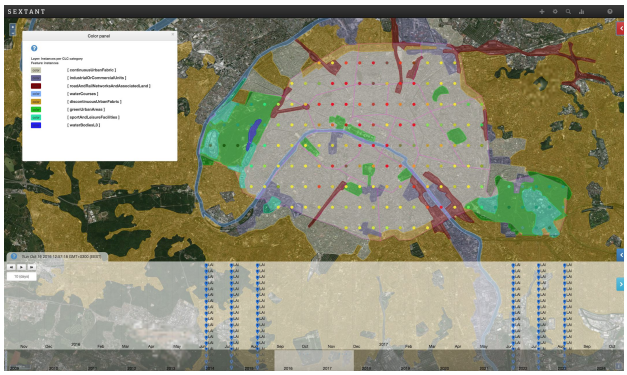

Figure 4: The "greenness" of Paris

All RDF datasets that have been discussed above are freely available at the following Web page: http://kr.di.uoa.gr/#datasets

The "greenness of Paris" case study can also be developed using the workflow on the right in the Copernicus App Lab software architecture of Figure 1. In this case, the datasets of interest can be queried using Ontop-spatial and visualized in Sextant without having to transform any datasets into RDF. In this case, the developer has to write R2RML mappings expressing the correspondence between a data source and classes/properties in the corresponding ontology. An example of such a mapping is provided in Listing 2 (in the native mapping language of Ontop-spatial which is less verbose than R2RML).

Listing 2: Example of mappings

```
mappingId opendap_mapping
target lai:{id} rdf:type lai:Observation .
       lai:{id} lai:lai {LAI}^^xsd:float;
                time:hasTime {ts}^^xsd:dateTime .
                lai:{id} geo:hasGeometry _:g .
                _:g geo:asWKT {loc}^^geo:wktLiteral .
source SELECT id, LAI, ts, loc
       FROM (ordered opendap
      url:https://analytics.ramani.ujuizi.com/
      thredds/dodsC/Copernicus-Land-timeseries-
      global-LAI%29/readdods/LAI/, 10)
       WHERE LAI > 0
```

In the example mappings provided in 2, the `source` is the LAI dataset discussed above which provided through the RAMANI OPeNDAP server of the Copernicus App Lab software stack. The dataset contains observations that are LAI values as well as the time and location for each observation. The MadIS operator `Opendap` retrieves this data and populates a virtual SQL table with schema `(id,LAI,ts,loc)`. The column `id` was not originally in the dataset but it is constructed from the location and the time of observation. The LAI column stores LAI values of an observation as `float` values. The attribute `ts` represents the timestamp of an observation in date-time format. In the original dataset times are given as numeric values and their meaning is explained in the metadata. For example, it can be days or months after a certain time origin. The `Opendap` virtual table operator converts these values to a standard format. Because of the fact that the `Opendap` operator is implemented as an SQL user-defined operator, it can be embedded into any SQL query. In the above mapping, we also refine the data that we want to be translated into virtual RDF terms by adding an filter to the query to eliminate (noisy) negative or zero LAI values. The value `10` that is passed as argument to the `Opendap` virtual table operator is the length of the time window w of the cache that is used (in minutes). In this case, if $|w|$ is the length of the time window w, then $|w| = 10$ minutes. This means that results of a every OPeNDAP call get cached every 10 minutes. If a query arrives resulting in an OPeNDAP in time t, where $t < 10$ minutes later than a previous *identical* OPeNDAP call (resulting from a same or similar query that involves the same OPeNDAP call), then the cached results can be used directly, eliminating the cost of performing another call to the OPeNDAP server.

The `target` part of the mapping encodes how the relational data is mapped into RDF terms. Every row in the virtual table describes an instance of the class `lai:Observation` of the LAI ontology in Figure 2. The values of the LAI column populate the triples that describe the LAI values of the observation, and the values of the columns `ts` and `loc` populate the triples that describe the time and location of the observations accordingly.

Given the mapping provided above, we can pose the GeoSPARQL query provided in Listing 3 to retrieve the LAI values and the geometries of the corresponding areas.

Listing 3: Query retrieving LAI values and locations

```
SELECT DISTINCT ?s ?wkt ?lai
WHERE { ?s lai:hasLai ?lai .
        ?s geo:hasGeometry ?g .
        ?g geo:asWKT ?wkt }
```

Using queries like the one described in Listing 3, Sextant can again visualize the various datasets and build layered maps like the one in Figure 4. The visualization of the case study in Sextant is available on line at the following URL: http://test.strabon.di.uoa.gr/SextantOL3/?mapid=m8s4kilcarub1mun_

5 LESSONS LEARNED, FUTURE PLANS AND OPEN PROBLEMS

In this section, we discuss lessons learned, future plans and open problems in the following areas of research and development of the project: the cloud platform, the use of OPeNDAP for accessing global land service and PROBA-V data at VITO, the linked geospatial data technologies and the use of our technologies by users that are not experts in Earth Observation.

The Terradue cloud platform. The use of this platform has provided the Copernicus App Lab project with the ability to manage all software components as cloud appliances, manage releases of the project software stack, deploy on demand this software stack on target infrastructures (e.g., at VITO), monitor operations (by each partner for its part), provide a development and integration environment, and manage solution updates and transfer to operations via cloud bursting. In this way, when the five DIAS will be operational, the Copernicus App Lab software will also be able to run on them. To demonstrate this, we will be working closely with EUMETSAT and Mercator Ocean to make the Copernicus App Lab software run on their DIAS.

The AppLab Cloud solution is operated through both the AppLab Front-end service for Mobile App developers, and the AppLab Back-end service for partnerships with data providers. The concept of operation of the AppLab Cloud solution supports mobile App developers and provides an AppLab capacity "as-a-service" to them. Considering the strength of having several technology providers contributing to the AppLab architecture, the challenge is about how to streamline a deployment scenario that can be replicated on different Copernicus Collaborative Ground Segment (CCGS) partner environments. Each individual deployment must be configured for the target environment, in order to enable some specific data management functions. All together, these deployments also have to be configured on top of the CCGS data repositories, in order to deliver a "value adding" service, aimed at Mobile App developers, over the Copernicus Services data products. Terradue Cloud Platform supports this "AppLab service" integration work, based on the contributed Docker services and on standard protocols, and deploys it "as-a-Service" on a selected target environment. A pre-operational capacity, is providing managed services on Terradue Cloud Platform. For the back-end services, Terradue Cloud Platform supports the deployment (Cloud bursting) of tailored data access services onto the Cloud Computing layer of a selected CCGS provider. The deployment scenario relies on using Terradue Cloud Platform as a reference platform for evolutive maintenance and versioning of the whole AppLab solution, and its updates as new deployments on the selected data providers' environments (e.g. a Copernicus DIAS).

Deploying OPeNDAP at VITO. The use of OPeNDAP offers better data access capabilities specifically for application developers that are not experts in Earth Observation, and thus it a clear benefit. OPeNDAP and SDL provide streaming data to the end user and have some significant advantages over the OGC Web Coverage Service standard which is already offered by VITO. First of all, from a data provider perspective, OPeNDAP is easier to use, as it is able to deal with a wider variety of grid types. Furthermore, OPeNDAP can be easily extended with different conventions, allowing for easier integration of different dataset and without overhead like file conversion. Also, OPeNDAP enables the loose coupling of different Copernicus data sources into one data model, providing the user easy access through a single access point that uses this data model. Finally, when using the Web Coverage Service, there is limited possibility to obtain client-specific parts of the datasets (one is limited to, for example, a bounding-box). In contrast, OPeNDAP allows for the caching of datasets by serialization based on internal array indices. This increases cache-hits for recurrent requests of a specific subpart of the dataset which can be very useful, e.g., in a mobile application scenario, where the viewport of the application could be defaulting to a specific, user-configurable area of interest with only modest panning and zooming interaction. Also, OPeNDAP ensures that metadata is intrinsically embedded in the TCP/IP response, regardless of container type (GeoTIFF, NetCDF, HDF, grib, etc.), which is beneficial for the semantic enrichment process that may happen at higher layers of the architecture. For the remaining duration of the project, the OPeNDAP deployment at VITO will be improved by offering a more sophisticated access control facility.

In order to deploy DAP at VITO we created several Virtual Machines (VMs) to build a private cloud in the VITO data center and used Docker images for the configuration. To provide access to the Copernicus and Proba-V datasets via the DAP, these datasets are mounted on the virtual machine. The data available on these storage volumes will be exposed by the DAP. Copernicus data uses the netCDF extension while tiff extension is used for Proba-V data. During the implementation of the DAP at VITO, it became clear that the directory structure used for the Copernicus Global Land datasets is not supported by the DAP, so we had to create a virtual directory structure to be compliant with the DAP. The reason why the Copernicus dataset could not be supported is that this directory structure contains multiple versions of data for the same day: the production centre reprocesses data at several days when more accurate meteorological data becomes available. The DAP could not handle this deviation, so VITO made a script to create a directory structure that uses symbolic links to point at the most recent version of the data, as that is the only one that needs to be exposed. Also, to ensure security we used tokens that allow accessing the datasets throught the RAMANI API. Every user has to register an account on the RAMANI platform. Without proper registration users will not have any access to the datasets to ensure map uptake monitoring capabilities and to avoid abuse. Furthermore, this will allow the tracking of which users access which datasets.

The linked geospatial data tools. The linked geospatial data tools presented in Section 3 have been used to develop environmental applications not only in Copernicus App Lab but also in previous projects TELEIOS, LEO and MELODIES [14, 17]. The linked geospatial data tools have been welcome by users who could see the value of developing applications using semantic technologies. The most popular tools have been Sextant and Ontop-spatial. Sextant has been irreplaceable as there is currently no other tool for visualizing linked geospatial data. Ontop-spatial has been attractive for users given that most of them were not in favour of transforming their data into RDF and storing it in Strabon. The fact that Ontop-spatial is also faster than Strabon on most of the queries of the benchmark Geographica [4] has been another reason users preferred it over Strabon.

The most innovative, but also challenging, aspect of using Ontop-spatial in Copernicus App Lab has been its ability to give access to Copernicus data (e.g., LAI data) through the OPeNDAP framework. When the data gets downloaded at query-time, query

execution typically takes two orders of magnitude more time than in the case where the data is materialized in a database or an RDF store. When data is stored in a database connected with Ontop-spatial, DBMS optimizations and database constraints are taken into account and query plans are optimized. This does not happen in the case where Ontop-spatial retrieves data on-the-fly from OPeNDAP, especially since data is preprocessed before it gets translated into virtual triples using Ontop-spatial. However, in the cases when we want to access Copernicus data that gets frequently updated, the virtual RDF graphs approach is useful as it avoids the repeated translation steps that have to be done by the data provider. For more costly operations (e.g., spatial joins of complex geometries), it is better to materialize the data. In our current work, we are developing further optimizations to improve the performance of this mechanism using techniques such as caching. We are also working on enabling Ontop-spatial to query other kinds of data e.g., HTML tables and social media data, since this functionality has been requested by users.

The following open problems will also be considered in the future:

- Although Strabon has been shown to be the most efficient spatiotemporal RDF store available today[37], much remains to be done for Strabon to scale to the *petabytes* of Copernicus data or the linked geospatial data typically managed by a national cartographic agency (e.g., Ordnance Survey in the United Kingdom or Kadaster in the Netherlands; they both use linked geospatial data). We plan to extend a scalable RDF store like Apache Rya[38] with GeoSPARQL support taking into account the lessons learned by similar projects in the relational world [13].
- It will usually be the case that different geospatial RDF datasets (e.g., GADM and OpenStreetMap) will be offered by different GeoSPARQL endpoints that can be considered a federation. There is currently no query engine that can answer GeoSPARQL queries over such a federation. The only system that comes close is SemaGrow which has been shown to federate a single Virtuoso endpoint and a single Strabon endpoint in [12]; more research is needed in this area for developing a state-of-the-art federation engine for GeoSPARQL.
- It is important to extend GeoTriples to be able to transform data in SQLite and ESRI geodatabases into RDF. The same should be done for scientific data formats such as NetCDF. The problem of representing array/gridded data in RDF has recently received attention by the W3C/OGC Spatial Data on the Web Working Group and some interesting working group notes were produced[39]. In a similar spirit an extension of SPARQL, called SciSPARQL, for querying scientific data has been proposed in the Ph.D. thesis of [1].

Using the Copernicus App Lab tools. Participants of the ESA Space App Camps that were organised in September 2017 and 2018 had the opportunity to use the Copernicus App Lab technologies to implement demo applications. ESA Space App Camps[40] are yearly events that bring together programmers to develop innovative applications based on satellite data. The objective is to make EO data, particularly from Copernicus, accessible to a wide range of businesses and citizens. Twenty-four developers from 14 countries attended the 2017 ESA App Camp in Frascati, Italy. It is important to note that these were the first two ESA Space App Camps that introduced the Copernicus App Lab tools and in both competitions he winning applications utilized the Copernicus App Lab technologies. The winning app of 2017 named AiR, displays an interactive projection of the Earth's surface to airplane travelers using Copernicus satellite imagery, letting them see information about the cities and landmarks they pass over during their flight, without the disruptions of clouds or parts of the plane getting in the way. The developers of AiR used Copernicus App Lab tools to access and integrate data from different sources (Copernicus land monitoring service data, OpenStreetMap data and DBpedia data about landmarks). In 2018 the winning app named Urbansat aims to guide greener, more ecological urban planning. It provides a range of data for planners, including information on green spaces, terrain and biodiversity and more. The app's map interface has a drag and drop feature, which would allow users to compare scenarios pre and post build for their construction projects, through the generation of relevant data, largely derived from Sentinel satellites. This would allow them to see the anticipated impact of constructing a building on local air quality, for example. The developers of Urbansat used Copernicus App Lab tools to process data from different sources (Copernicus land monitoring service data, Urban Atlas data, Natura 2000 data and data provided from GADM). Another interesting application that was developed was Track Champ. Track Champ combines Earth observation data with data about points of interest from OpenStreetMap to find the perfect time and place to exercise while tracking personal performance over time.[41]

One of the lessons that we learned from the 2017 and 2018 ESA Space App Camps, after compiling user feedback, was that the capability of integrating Earth observation data together with linked open data (e.g., landmarks, points of interest) was very important in some applications. In all of the applications that were based on combining Copernicus data with open data, developers chose to use the Copernicus App Lab tools. Given the limited time they had available for coding, they found it easier to access the available GeoSPARQL endpoints rather than the original heterogeneous data sources. On the other hand, some developers also reported that getting familiarized with the Copernicus App Lab technologies required some effort, given that they did not have background knowledge in Semantic Web technologies or geospatial data management. These users suggested that this issue could be addressed by improving documentation material.

Google has recently activated the beta version of its dataset search (https://toolbox.google.com/datasetsearch), where the datasets that are indexed using schema.org (https://schema.org/), as proposed by Google, show up. Schema.org is a vocabulary created from the collaboration of four major search engines: Bing, Google, Yahoo, and Yandex. It aims to provide a unique structured data markup schema which would include a great amount of topics, including people, organizations, events, creative works such as books and movies, etc. The on-page markup allows search engines to understand information included in web pages, while it provides rich search features for users. We have followed these guidelines and annotated all the datasets used in the use case of Section 3, and made them available at the

[37] We have shown this in papers [6, 15] in 2013. The situation has not changed since then, according to recent experiments of our team.
[38] https://rya.apache.org/
[39] https://www.w3.org/2015/spatial/wiki/Main_Page
[40] http://www.app-camp.eu/
[41] All the applications developed are presented in more detail at the following Web page: https://www.app-camp.eu/winner-frascati-2017/

following link: http://kr.di.uoa.gr/#datasets. We have also recommended that the same practice is followed by the Copernicus services we have worked with (land monitoring, global land and atmosphere services).

Currently, EO datasets are hidden in the archives of big EO organizations, such as ESA, NASA, etc. These datasets are only available through specialized search interfaces provided by the organizations. It is important to make major search engines like Google able to discover EO datasets and not only general datasets, in the same way they can discover information about movies, concerts, etc. To achieve this goal, we designed an extension to the community vocabulary schema.org, appropriate for annotating EO data in general and Copernicus data in particular, by extending the class Dataset with subclasses and properties, which cover the EO dataset metadata defined in the specification OGC 17-003 for annotating EO product metadata using GeoJSON(-LD) (http://geo.spacebel.be/opensearch/myDocumentation/doc/index-en.html). The OGC 17-003 is based on the specifications OCC 10-157r4 (http://docs.opengeospatial.org/is/10-157r4/10-157r4.html) and UMM-G (https://wiki.earthdata.nasa.gov/display/CMR/CMR+Documents), and it is expected to be standardized by OGC during 2018. The OGC 10-157r4 - Earth Observation Profile of Observations and Measurements (O&M) provides a standard schema for encoding EO product metadata in order to describe and catalogue products from the sensors of EO satellites. The Unified Metadata Model for Grabules (UMM-G) is an extensible metadata model for Granules, that provides the mappings between NASA's Common Metadata Repository (CMR) supported metadata standards.

The schema.org extension for encoding EO metadata can be used by EO organizations like ESA for the encoding of the metadata of their EO datasets. In this way, EO datasets will be discoverable by search engines. In addition, this schema.org extension for EO products can be used by webmasters who want to annotate their webpages, so that search engines can find the EO datasets they provide. It is important to make this EO data available on the Web as linked data in order to increase their use by developers that might not be experts in EO. In this way, great amounts of data that are generated fast, can be made "interoperable" and more valuable when they are linked together.

6 SUMMARY

In this paper we argued that Copernicus data is a paradigmatic source of big data giving rise to all relevant challenges: volume, velocity, variety, veracity and value. The Copernicus App Lab project targets all these challenges with special focus in variety and volume, and has developed a novel software stack that can be used to develop applications using Copernicus data even by developers that are not experts in Earth observation. We presented a case study developed using the Copernicus App Lab software stack and discussed lessons learned, future plans and open problems.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825258 and Greek national funds through the Operational Program "Competitiveness, Entrepreneurship and Innovation", under the call "RESEARCH-CREATE-INNOVATE" (project code: T1EDK01000).

REFERENCES

[1] A. Andrejev. 2016. *Semantic Web Queries over Scientific Data.* Ph.D. Dissertation. Dept. of Information Technology, Uppsala University, Sweden.
[2] K. Bereta, H. Caumont, U. Daniels, D. Dirk, M. Koubarakis, D. Pantazi, G. Stamoulis, S. Ubels, V. Venus, and F. Wahyudi. 2019. From Big Copernicus Data to Big Information and Big Knowledge: the Copernicus App Lab project. In *BiDS*.
[3] K. Bereta, H. Caumont, E. Goor, M. Koubarakis, D.-A. Pantazi, G. Stamoulis, S. Ubels, V. Venus, and F. Wahyudi. 2018. From Copernicus Big Data to Big Information and Big Knowledge: A Demo from the Copernicus App Lab Project. In *CIKM*.
[4] K. Bereta and M. Koubarakis. 2016. Ontop of Geospatial Databases. In *ISWC*.
[5] K. Bereta and M. Koubarakis. 2017. Creating Virtual Semantic Graphs ontop of Big Data from Space. In *BiDS*.
[6] K. Bereta, P. Smeros, and M. Koubarakis. 2013. Representation and Querying of Valid Time of Triples in Linked Geospatial Data. In *ESWC*.
[7] J. Blower, M. Riechert, N. Pace, and M. Koubarakis. 2016. Big Data meets Linked Data: What are the Opportunities?. In *Conference on Big Data from Space (BiDS), Tenerife, Spain, March 2016.*
[8] S. Brüggemann, K. Bereta, G. Xiao, and M. Koubarakis. 2016. Ontology-Based Data Access for Maritime Security. In *ESWC*.
[9] S. Burgstaller, W. Angermair, F. Niggemann, S. Migdall, H. Bach, I. Vlahopoulos, D. Savva, P. Smeros, G. Stamoulis, K. Bereta, and M. Koubarakis. 2017. LEOpatra: A Mobile Application for Smart Fertilization Based on Linked Data. In *Proceedings of the 8th International Conference on Information and Communication Technologies in Agriculture, Food & Environment.*
[10] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, and G. Xiao. 2017. Ontop: Answering SPARQL queries over relational databases. *Semantic Web* (2017).
[11] Y. Chronis, Y. Foufoulas, V. Nikolopoulos, A. Papadopoulos, L. Stamatogiannakis, C. Svingos, and Y. E. Ioannidis. 2016. A Relational Approach to Complex Dataflows. In *EDBT/ICDT*.
[12] A. Davvetas, I. A. Klampanos, S. Andronopoulos, G. Mouchakis, S. Konstantopoulos, A. Ikonomopoulos, and V. Karkaletsis. 2017. Big Data Processing and Semantic Web Technologies for Decision Making in Hazardous Substance Dispersion Emergencies. In *ISWC*.
[13] A. Eldawy and M. F. Mokbel. 2017. The Era of Big Spatial Data. VLDB.
[14] M. Koubarakis et al. 2016. Managing Big, Linked, and Open Earth-Observation Data: Using the TELEIOS/LEO software stack. *IEEE Geoscience and Remote Sensing Magazine* (2016).
[15] G. Garbis, K. Kyzirakos, and M. Koubarakis. 2013. Geographica: A Benchmark for Geospatial RDF Stores (Long Version). In *ISWC*.
[16] Manolis Koubarakis, Konstantina Bereta, George Papadakis, Dimitrianos Savva, and George Stamoulis. 2017. Big, Linked Geospatial Data and Its Applications in Earth Observation. *IEEE Internet Computing* 21, 4 (2017), 87–91. https://doi.org/10.1109/MIC.2017.2911438
[17] M. Koubarakis, K. Bereta, G. Papadakis, D. Savva, and G. Stamoulis. 2017. Big, Linked Geospatial Data and Its Applications in Earth Observation. *IEEE Internet Computing* (2017).
[18] M. Koubarakis, C. Kontoes, and S. Manegold. 2013. Real-time wildfire monitoring using scientific database and linked data technologies. In *EDBT*.
[19] M. Koubarakis, K. Kyzirakos, C. Nikolaou, G. Garbis, K. Bereta, R. Dogani, S. Giannakopoulou, P. Smeros, D. Savva, G. Stamoulis, G. Vlachopoulos, S. Manegold, C. Kontoes, T. Herekakis, I. Papoutsis, and D. Michail. 2016. Managing Big, Linked, and Open Earth-Observation Data Using the TELEIOS/LEO software stack. *IEEE Geoscience and Remote Sensing Magazine* (2016).
[20] K. Kyzirakos, M. Karpathiotakis, G. Garbis, C. Nikolaou, K. Bereta, I. Papoutsis, T. Herekakis, D. Michail, M. Koubarakis, and C. Kontoes. 2014. Wildfire monitoring using satellite images, ontologies and linked geospatial data. *J. Web Sem.* (2014).
[21] K. Kyzirakos, M. Karpathiotakis, and M. Koubarakis. 2012. Strabon: A Semantic Geospatial DBMS. In *ISWC*.
[22] K. Kyzirakos, D. Savva, I. Vlachopoulos, A. Vasileiou, N. Karalis, M. Koubarakis, and S. Manegold. 2018. GeoTriples: Transforming Geospatial Data into RDF Graphs Using R2RML and RML Mappings. *Journal of Web Semantics* (2018).
[23] K. Kyzirakos, I. Vlachopoulos, D. Savva, S. Manegold, and M. Koubarakis. 2014. GeoTriples: a Tool for Publishing Geospatial Data as RDF Graphs Using R2RML Mappings. In *Proc. of TerraCognita.*
[24] C. Nikolaou, K. Dogani, K. Bereta, G. Garbis, M. Karpathiotakis, K. Kyzirakos, and M. Koubarakis. 2015. Sextant: Visualizing time-evolving linked geospatial data. *Journal of Web Semantics* (2015).
[25] G. Papadakis, K. Bereta, T. Palpanas, and M. Koubarakis. 2017. Multi-core Meta-blocking for Big Linked Data. In *SEMANTICS.*
[26] K. Patroumpas, N. Georgomanolis, T. Stratiotis, M. Alexakis, and S. Athanasiou. 2015. Exposing INSPIRE on the Semantic Web. *Web Semant.* (2015).
[27] M. Perry and J. Herring. 2012. GeoSPARQL - A geographic query language for RDF data. OGC.
[28] P. Smeros and M. Koubarakis. 2016. Discovering Spatial and Temporal Links among RDF Data. In *LDOW*.
[29] C. Stadler, J. Lehmann, K. Höffner, and S. Auer. 2012. LinkedGeoData: A core for a web of spatial open data. *Semantic Web* (2012).