Piveau: A Large-scale Open Data Management Platform based on Semantic Web Technologies

Fabian Kirstein[1,2] ✉, Kyriakos Stefanidis[1] Benjamin Dittwald[1], Simon Dutkowski[1], Sebastian Urbanek[1,2], and Manfred Hauswirth[1,2,3]

[1] Fraunhofer FOKUS, Berlin, Germany
[2] Weizenbaum Institute for the Networked Society, Berlin, Germany
[3] TU Berlin, Open Distributed Systems, Berlin, Germany
{firstname.lastname}@fokus.fraunhofer.de

Abstract. The publication and (re)utilization of Open Data is still facing multiple barriers on technical, organizational and legal levels. This includes limitations in interfaces, search capabilities, provision of quality information and the lack of definite standards and implementation guidelines. Many Semantic Web specifications and technologies are specifically designed to address the publication of data on the web. In addition, many official publication bodies encourage and foster the development of Open Data standards based on Semantic Web principles. However, no existing solution for managing Open Data takes full advantage of these possibilities and benefits. In this paper, we present our solution "Piveau", a fully-fledged Open Data management solution, based on Semantic Web technologies. It harnesses a variety of standards, like RDF, DCAT, DQV, and SKOS, to overcome the barriers in Open Data publication. The solution puts a strong focus on assuring data quality and scalability. We give a detailed description of the underlying, highly scalable, service-oriented architecture, how we integrated the aforementioned standards, and used a triplestore as our primary database. We have evaluated our work in a comprehensive feature comparison to established solutions and through a practical application in a production environment, the European Data Portal. Our solution is available as Open Source.

Keywords: Open Data · DCAT · Scalability.

1 Introduction

Open Data constitutes a prospering and continuously evolving concept. At the very core, this includes the publication and re-utilization of datasets. Typical actors and publishers are public administrations, research institutes, and non-profit organizations. Common users are data journalists, businesses, and governments. The established method of distributing Open Data is via a web platform that is responsible for gathering, storing, and publishing the data. Several software solutions and specifications exist for implementing such platforms. Especially the Resource Description Framework (RDF) data model and its associated vocabularies represent a foundation for fostering interoperability and harmonization of different data sources. The Data Catalog Vocabulary (DCAT) is applied as a comprehensive model and standard for describing datasets and data services on Open Data platforms [1]. However, RDF is only a subset of the Semantic Web stack and Open Data publishing does not benefit from the stack's full potential, which offers more features beyond data modeling. Therefore, we developed a novel and scalable platform for managing Open Data, where the Semantic Web stack is a first-class citizen. Our work focuses on two central aspects: (1) The utilization of a variety of Semantic Web standards and technologies for covering the entire life-cycle of the Open Data publishing process. This covers particularly data models for metadata, quality verification, reporting, harmonization, and machine-readable interfaces. (2) The application of state-of-the-art software engineering approaches for development and deployment to ensure production-grade applicability and scalability. Hence, we integrated a tailored microservice-based architecture and a suitable orchestration pattern to fit the requirements in an Open Data platform.

It is important to note, that currently our work emphasizes the management of metadata, as intended by the DCAT specification. Hence, throughout the paper the notion of data is used in terms of metadata.

In Section 2 we describe the overall problem and in Section 3 we discuss related and existing solutions. Our software architecture and orchestration approach is described in Section 4. Section 5 gives a detailed overview of the data workflow and the applied Semantic Web standards. We evaluate our work in Section 6 with a feature analysis and an extensive use case. To conclude, we summarize our work and give an outlook for future developments.

2 Problem Statement

A wide adoption of Open Data by data providers and data users is still facing many barriers. Beno et al. [7] conducted a comprehensive study of these barriers, considering legal, organizational, technical, strategic, and usability aspects. Major technical issues for users are the limitations in the Application Programming Interfaces (APIs), difficulties in searching and browsing, missing information about data quality, and language barriers. Generally, low data quality is also a fundamental issue, especially because (meta)data is not machine-readable or, in many cases, incomplete. In addition, low responsiveness and bad performance of the portals have a negative impact on the adoption of Open Data. For publishers, securing the integrity and authenticity, enabling resource-efficient provision, and clear licensing are highly important issues. The lack of a definite standard and technical solutions is listed as a core barrier.

The hypothesis of our work is, that **a more sophisticated application of Semantic Web technologies can lower many barriers in Open Data publishing and reuse**. These technologies intrinsically offer many aspects, which are required to improve the current support of Open Data. Essentially, the Semantic Web is about defining a common standard for integrating and harnessing data from heterogeneous sources [2]. Thus, it constitutes an excellent

match for the decentralized and heterogeneous nature of Open Data.
Widespread solutions for implementing Open Data platforms are based on canonical software stacks for web applications with relational and/or document databases. The most popular example is the Open Source solution Comprehensive Knowledge Archive Network (CKAN) [10], which is based on a flat JSON data schema, stored in a PostgreSQL database. This impedes a full adoption of Semantic Web principles. The expressiveness of such a data model is limited and not suited for a straightforward integration of RDF.

3 Related Work

Making Open Data and Linked Data publicly available and accessible is an ongoing process that involves innovation and standardization efforts in various topics such as semantic interoperability, data and metadata quality, standardization as well as toolchain and platform development.

One of the most widely adopted standards for the description of datasets is DCAT and its extension DCAT Application profile for data portals in Europe (DCAT-AP) [12]. The latter adds metadata fields and mandatory property ranges, making it suitable for use with Open Data management platforms. Its adoption by various European countries led to the development of country-specific extensions such as the official exchange standard for open governmental data in Germany [17] and Belgium's extension [24]. Regarding Open Data management platforms, the most widely known Open Source solution is CKAN [10]. It is considered the de-facto standard for the public sector and is also used by private organizations. It does not provide native Linked Data capabilities but only a mapping between existing data structures and RDF. Another widely adopted platform is uData [23]. It is a catalog application for collecting data and metadata focused on being more contributive and inclusive than other Open Data platforms by providing additional functionality for data reuse and community contributions. Other Open Source alternatives include the repository solution DSpace which dynamically translates [13] relational metadata into native RDF metadata and offers it via a SPARQL endpoint. WikiData also follows a similar approach [36]; it uses a custom structure for identifiable items, converts them to native RDF and provides an API endpoint. Another, proprietary, solution is OpenDataSoft [26], which has limited support for Linked Data via its interoperability mode. There are also solutions that offer native Linked Data support following the W3C recommendation for Linked Data Platforms (LDPs). Apache Marmotta [38] has native implementation of RDF with a pluggable triplestore for Linked Data publication. Virtuoso [27] is a highly scalable LDP implementation that supports a wide array of data access standards and output formats. Fedora [21] is a native Linked Data repository suited for digital libraries. Recent research efforts [30] focusses on the notion of dynamic Linked Data where context aware services and applications are able to detect changes in data by means of publish-subscribe mechanisms using SPARQL.

A core feature of most big commercial platforms is the Extract, Transform, Load (ETL) functionality. It refers to the three basic data processing stages of reading data (extract) from heterogeneous sources, converting it (transform) to a suitable format, and storing it (load) into a database. Platforms that offer ETL as a core functionality include IBM InfoSphere [16] with its DataStage module, Oracle Autonomus Data Warehouse [28] with its Data Integrator module and SAS Institute's data warehouse [31]. Moreover, various Open Source solutions such as Scriptella [35] and Talend Open Studio [32] are based on ETL. The above data warehouses offer highly scalable ETL functionality but do not support Linked Data and DCAT. On the other hand, the previously mentioned Linked Data platforms do not offer any real ETL capabilities. Bridging this gap was the main objective that led to the development of the Piveau pipeline as a core part of our architecture. Similar data pipelines can be found as stand-alone services and applications such as AWS Data Pipeline [5], Data Pipes from OKFN [25], North Concepts Data Pipeline [22], and Apache Airflow [33].

4 A Flexible Architecture for Semantic Web Applications

Semantic Web technologies are mainly supported by specifications, standards, libraries, full frameworks, and software. The underlying concept of our architecture is the encapsulation of Semantic Web functionalities to make them reusable and interoperable, which is considered a classical software engineering principle. Our Open Data platform introduces a state-of-the-art, tailored architecture to orchestrate these encapsulations and make them easy to apply in production environments. It is based on a microservice architecture and a custom pipeline system, facilitating a flexible and scalable feature composition of Open Data platforms. This enables the application of Piveau for various use cases and audiences. Furthermore, it enables the re-use of features in other environments and applications.

4.1 The Piveau Pipeline

The basic requirements of our architecture were the use of microservices, high scalability, lightweight in application and management, and suitable for large-scale data processing. Existing workflow engines and ETL systems are either not designed for Linked Data and/or limited solely to extensive data integration tasks (see Section 3). To lower complexity and maintenance needs, we aimed for an unifying architecture and data processing concept, which targets specifically our needs. Therefore, we designed and implemented the Piveau pipeline (PPL). The PPL builds upon three principal design choices: (1) All services and features expose RESTful interfaces and comply with the microservice style. (2) The services can be connected and orchestrated in a generic fashion to implement specific data processing chains. (3) There is no central instance, which is responsible for orchestrating the services.

A PPL orchestration is described by a *descriptor*, which is a plain JSON document, including a list of segments, where each segment describes a step (a service) in the data processing chain. Every segment includes at least meta-information, targeting the respective service and defining the consecutive service(s).[4] The entire descriptor is passed from service to service as state information. Each service identifies its segment by a service identifier, executes its defined task and passes the descriptor to the next service(s). Hence, the descriptor is a compilation and self-contained description of a data processing chain. Each microservice must expose an endpoint to receive the descriptor and must be able to parse and execute its content. The processed data itself can be embedded directly into the descriptor or passed via a pointer to a separate data store, e.g. a database, file system or other storage. This depends on the requirements and size of data and can be mixed within the process.

The PPL has been proven to be a fitting middle ground between ETL approaches and workflow engines. On an architectural level, it allows to harvest data from diverse data providers and orchestrate a multitude of services. Its production-level implementation in the European Data Portal (EDP) supports millions of open datasets with tens of thousands updates per day (see Section 6.2).

4.2 Architecture, Stack and Deployment

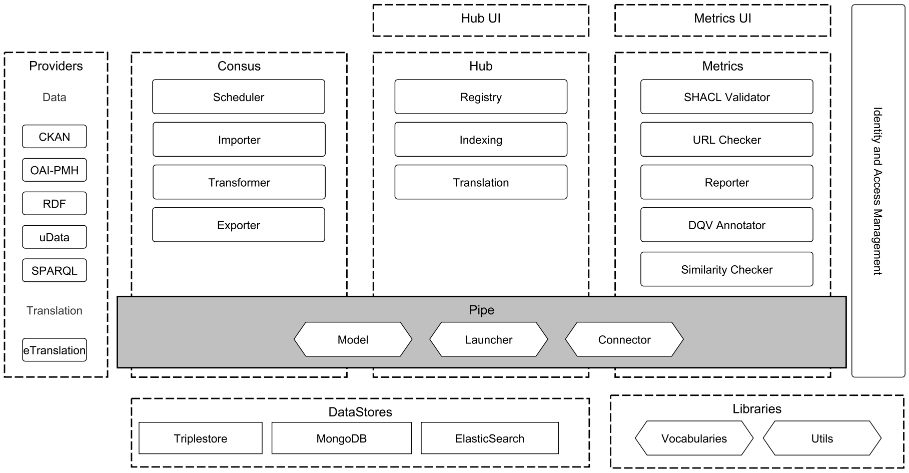

Fig. 1. Piveau High-Level Architecture

[4] The PPL descriptor schema can be found at: `https://gitlab.com/piveau/pipeline/piveau-pipe-model/-/blob/master/src/main/resources/piveau-pipe.schema.json`

The development of Piveau follows the reactive manifesto, which requires a system to be responsive, resilient, elastic, and message driven [9]. The platform is divided into three logical main components, each one responsible for a phase within the life-cycle of the datasets: Consus, Hub and Metrics. Figure 1 illustrates the overall architecture and structure.

Consus is responsible for the data acquisition from various sources and data providers. This includes scheduling, transformation and harmonization. Hub is the central component to store and register the data. Its persistence layer consists of a Virtuoso triplestore[5] as the principal database, Elasticsearch[6] as the indexing server and a MongoDB[7] for storing binary files. Metrics is responsible for creating and maintaining comprehensive quality information and feeding them back to the Hub. Two web applications based on Vue.js[8] are available for browsing the data. The services are written with the reactive JVM framework Vert.x[9] and orchestrated with the PPL within and across the logical components. Several libraries for common tasks, RDF handling and the PPL orchestration are re-used in all services.
In order to enable native cloud deployment, we use the Docker[10] container technology. Each service is packaged as a container, supporting easy and scalable deployment. In addition, Piveau was tested with Kubernetes-based[11] container management solutions like Rancher[12] and OpenShift[13]. Hence, our architecture supports a production-grade development scheme and is ready for DevOps practices.

4.3 Security Architecture

In this section we will describe how Piveau handles authentication, authorization, and identity management. The multitude of standardized system and network security aspects that are part of the Piveau architectural design, such as communication encryption, firewall zones and API design, are beyond the scope of this paper.

Piveau is comprised of multiple microservices, Open Source software and a set of distinct web-based user interfaces. In order to support Single Sing-On (SSO) for all user interfaces and authentication/authorization to all microservices, we use Keycloak[14] as central identity and access management service. Keycloak also supports federated identities from external providers. Specifically, in the case of the EDP, we use ”EU Login” as the sole external identity provider without

[5] `https://virtuoso.openlinksw.com/`
[6] `https://www.elastic.co/products/elasticsearch`
[7] `https://www.mongodb.com/`
[8] `https://vuejs.org/`
[9] `https://vertx.io/`
[10] `https://www.docker.com/`
[11] `https://kubernetes.io/`
[12] `https://rancher.com/`
[13] `https://www.openshift.com/`
[14] `https://www.keycloak.org/`

allowing any internal users apart from the administrators. Authentication and authorization on both front-end and back-end services follows the OIDC protocol [34]. More specifically, all web-based user interfaces follow the OIDC authorization code flow. This means that when a user tries to login to any of Piveau's user interfaces, they are redirected to the central Keycloak authentication form (or the main identity provider's authentication form) and, upon successful login, they are redirected back to the requested web page. This provides a uniform user experience and minimizes the risk of insecure implementation of custom login forms.

All back-end services also follow OIDC by requiring valid access tokens for each API call. Those tokens follow the JSON Web Token (JWT) standard. In contrast to static internal API keys, this design pattern supports arbitrary back-end services to be open to the public without any change to their authentication mechanisms. Moreover, since the JWT tokens are self-contained, i.e. they contain all the required information for user authentication and resource authorization, the back-end services can perform the required checks without the need of communication with a database or Keycloak. Not requiring round-trips greatly enhances the performance of the whole platform.

The fine-grained authorization follows the User-Managed Access (UMA) specification [18], where resource servers (back-end services) and a UMA-enabled authorization server (Keycloak) can provide uniform management features to user-owned resources such as catalogs and datasets.

5 Semantic Data Workflow

In the following, a typical data flow in our Open Data platform is described to illustrate our solution in detail. This covers the process of acquiring the data from the original providers, evaluating the quality of that data, and presenting and managing the data (see Figure 2). We focus on the used Semantic Web technologies and specifications. The presented order reflects roughly the order of execution. But since many processes run asynchronously, the order can vary depending on their execution time.

5.1 Data Acquisition

The main entry point for any data workflow and orchestration is the **scheduler**. Each data workflow, defined as a PPL descriptor (see Section 4.1), is assigned a list of triggers. A trigger may define a periodical execution (hourly, daily, weekly, bi-weekly, yearly, etc.), number of execution times, a list of specific date and times to execute, or an immediate execution. Each trigger is able to pass its own process configuration in order to individualize the workflow depending on the execution time. Upon execution, the scheduler passes the descriptor to the first service in line, typically an **importer**.
An importer retrieves the metadata from the source portal(s). We have implemented a range of importers to support a variety of interfaces and data formats,

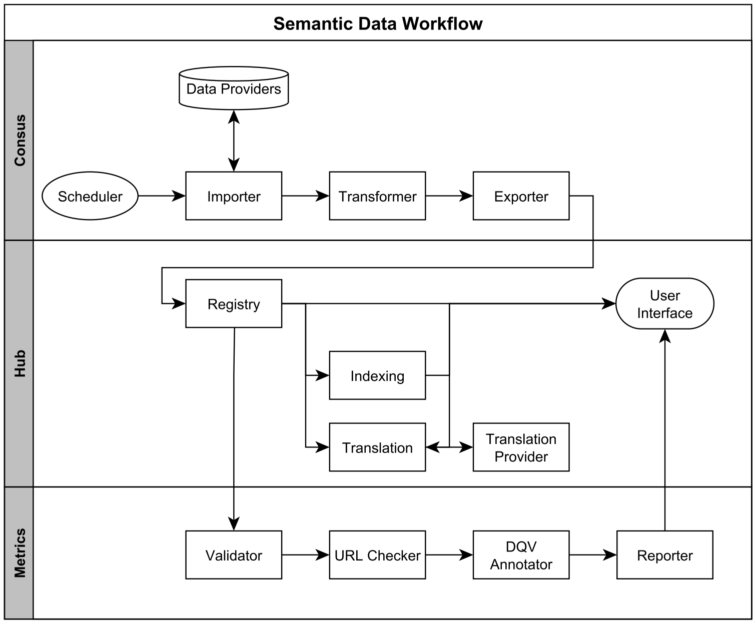

Fig. 2. Semantic Data Workflow

e.g. CKAN-API, OAI-PMH, uData, RDF, and SPARQL. The importer is responsible for extracting records of metadata from either an API or a dump file and for sending it to the next processing step. This covers the generation of a complete list of identifiers of all datasets, which will be required for a final synchronization, including the deletion of datasets, which are not present in the source portal anymore.

The principal data format of Piveau is RDF, therefore non-RDF or not supported RDF dialects sources require a transformation. A **transformer** generates RDF from such source data, by applying light-weight transformation scripts written in JavaScript. The final output is always DCAT-compliant RDF. The scripts can be managed externally (e.g. in Git) to ensure maintainability.

Finally, our **exporter** sends the RDF data to the Hub component. Non-existing datasets are deleted by the exporter based on the identifier list that is acquired in the importing step.

5.2 Processing and Storing

The central service for dataset management is the **registry**. It acts as a middleware and abstraction layer to interact with the triplestore. It offers a RESTful interface, supporting the major RDF serializations (Turtle, JSON-LD, N-Triples, RDF/XML, Notation3). Its resources reflect the main DCAT entities: catalog, dataset, and distribution. The main task is to pre-process and harmonize the data received from the exporter. This includes the application of consistent and meaningful URI schemata [6], the generation of unique IDs, and the mapping to linked, existing entities. It ensures the integrity and traceability of the data in the triplestore. The **indexing** service is responsible for managing the high-performance search index. It receives the processed RDF data from the registry and flattens it into a plain JSON representation, which is suitable for indexing. Firstly, this is done by extracting relevant literals from the data, e.g. from properties like title and description. Secondly, linked resources are resolved and proper literals are extracted from the result (for instance by looking for *rdfs:label*). The service supports the use of existing and well-maintained vocabularies and ontologies for that purpose. Piveau ships with a selection of vocabularies, e.g. for human languages, licenses, and geolocations. The result of the search service constitutes one of the main access points to the data, because it is much more human-readable than native RDF.

The **translation** service manages the machine translation of literals into multiple languages. It represents a middleware to third-party translations services, bundling strings from multiple datasets to an integrated request. After completion the service stores the translation by applying the native multi-language features of RDF. As soon as a dataset is retrieved, the existing original languages are identified and added to the text information using a language tag inside the dataset. This labeling is based on ISO 639-1 language codes. In addition, metadata about the translation status are stored in the dataset, indicating when a translation was started and when it was completed. Translated text information are labeled with an extended language tag to differentiate them from the original text. It follows the schema *en-t-de-t0-abc* [11], where the target language is named first, followed by a *t* and the original language.

Finally, the data is accessible via multiple means. The triplestore exposes a SPARQL endpoint, which offers raw und direct access to the data. A RESTful API allows the access to the RDF serializations, provided by the registry and to the indexed serializations, provided by the search service. A web **user interface** offers access to end users and interacts directly with the RESTful API.

5.3 Quality Evaluation

In parallel with the main data processing steps, the data is processed by dedicated services to assess its quality. Semantic Web technologies offer mature tools and standards to conduct this task.

The **validator** provides a formal validation of each dataset. We apply the W3C Shapes Constraint Language (SHACL) [20], where a pre-defined set of rules is tested against a dataset. Currently the DCAT-AP SHACL rules [15] are included. The validation results include detailed information about issues and violations. This result covers the exact paths and reasons for the identified deficits. The applied rules can also be extended or replaced. In addition, the **URL checker** performs accessibility tests on each linked distribution (the actual data) and assesses its availability via HTTP status codes.

The **DQV annotator** [4] provides a qualitative assessment for each dataset. It is based on a custom metrics scheme, which is inspired by the FAIR principles [39]. The findability dimension refers to completeness of the metadata, e.g. whether keywords, geo data or time information are provided. Accessibility refers to the results from the URL checker. Interoperability is assessed by evaluating the format and type of data, which is referenced in a dataset (distribution). For instance, if the data is in a machine-readable and/or non-proprietary format. Reusability is mostly confirmed by checking the availability of licensing information. Beyond this FAIR evaluation, the similarity of a dataset to other datasets is calculated based on locality-sensitive hashing (LSH) algorithm.

The results of the validation and annotator services are summarized in a quality report and attached as RDF to the concerned dataset in the triplestore. This report uses a custom quality vocabulary, which applies the W3C Data Quality Vocabulary (DQV) and reflects our metric scheme. In addition, an aggregated report is attached to the respective catalog.

The **reporter** offers a variety of human-readable versions of the quality reports. It collects all data from the triplestore and renders visually appealing reports of the information. It supports PDF, XLS or ODS. In addition, a comprehensive web front-end is available, and is integrated into the front-end of the Hub component.

6 Evaluation

We have evaluated our work according to three quantitative and qualitative aspects. In Section 6.1 we compare Piveau with two well-known Open Data solutions. In Section 6.2 we describe a real-world application based on Piveau. Finally, in Section 6.3 we present an analysis of the impact of Semantic Web technologies on the perceived barriers of Open Data.

6.1 Feature Comparison with Open Data Solutions

No definite metric exists to specifically assess the technical performance of Open Data technologies and infrastructures. However, a lot of work and research was conducted in the field of requirements and evaluation modeling for Open Data. An extensive review covering a broad variety of dimensions (economical, organizational, ergonomic, etc.) is presented by Charalabidis et al. [3] This includes an overview of "Functional Requirements of an Open Data Infrastructure", which acts as the main basis for our feature matrix [3]. It is supplemented by indicators from the outcome of "Adapting IS [Information Systems] Success Model on Open Data Evaluation" [3]. Furthermore, we translated the W3C recommendation for best practices for publishing data on the web into additional indicators [37]. Finally, the matrix is complemented by custom indicators to reflect our experiences in designing and developing Open Data infrastructures. In the selection process we only focused on indicators, which were applicable to measurable technical aspects that reflect the overall objective of managing metadata. More

personal indicators, like ”The web pages look attractive”, were not considered. Still, this approach led to a large number of indicators (>50), which we semantically combined to generate a compact and meaningful feature matrix.[15]
We compared Piveau with the popular Open Data solutions CKAN and uData (see Section 3). The selection criteria were: (1) Must be freely available as Open Source software; (2) Must not be a cloud- or hosting-only solution; (3) Has a high rate of adoption and (4) Primarily targets public sector data. Table 1 shows the final feature matrix and the result of the evaluation. Each measure was rated with the following scale: 0 - not supported, 1 - partially supported, 2 - fully supported. An explanation is given for each rating, where required.

The overall result indicates that our solution can match with existing and established solutions and even reaches the highest score. Piveau offers strong features regarding searching and finding datasets and data provision. The comprehensive metadata is a great foundation for analyses and visualizations. Our features for quality assurance are unrivaled and we support the most scalable architecture. Yet, uData offers unique features for interaction and CKAN is very mature and industry-proven.

6.2 The European Data Portal

The EDP[16] is a central portal, publishing all metadata of Open Data provided by public authorities of the European Union (EU). It gathers the data from national Open Data portals and geographic information systems. It was initially launched in November 2015 by the European Commission (EC). Its design and development was driven by the DCAT-AP specification.

The EDP was one of the first implementations of the DCAT-AP specification. In order to comply with established Open Data publishing concepts, the first version was based on an extended CKAN with an additional layer for transforming and replicating all metadata into RDF. This setup required additional mechanisms to transform data and, thus, proved to be too complex and limited for the growing amounts of Open Data in Europe. [19] We successfully improved this first version with our solution Piveau. This successfully enrolled our solution in a large-scale production environment. Our translation middleware integrates the eTranslation Service of the EU Commission [29], enabling the provision of metadata in 25 European languages. As of December 2019 the EDP offers approximately one million DCAT datasets, in total consisting of more than 170 million RDF triples, fetched from more than 80 data providers. Open Data is considered to be a key building block of Europe’s data economy [14], indicating the practical relevance of our work.

[15] The exact provenance and creation process of the feature matrix is available as supplementary material: `https://zenodo.org/record/3571171`

[16] `https://www.europeandataportal.eu`

	Piveau		CKAN		uData	
Searching and Finding Data						
Support for data federation	2	*Native support through SPARQL*	1	*Indirect through harvesting*	1	*Indirect through harvesting*
Integration of controlled vocabularies	2	*Support for structured controlled vocabulary*	1	*Support for simple controlled vocabulary*	1	*Support for simple controlled vocabulary*
Filtering, sorting, structuring, browsing and ordering search results by diverse dimensions	2	*Application of search engine*	2	*Application of search engine*	2	*Application of search engine*
Offer a strong and interoperable API	2	*DCAT compliant REST*	2	*DCAT compliant REST*	2	*DCAT compliant REST*
Support multiple languages	2	*On interface and dataset level*	1	*Only on interface level*	2	*On interface and dataset level*
Linked Data interface	2	*SPARQL endpoint*	0		0	
Geo-Search	2	*Available*	2	*Available*	2	*Available*
Data Provision and Processing						
Data Upload	1	*Binary data upload*	2	*Binary and structured data upload*	1	*Binary data upload*
Data Enrichment and Cleansing	0		0		0	
Support for linking and referring other data	2	*Any number of links possible*	1	*Restrictive schema*	1	*Restrictive schema*
Analysis and Visualization						
Provide comprehensive metadata	2	*Complete and extensible schema*	1	*Restricted schema*	1	*Restricted schema*
Offer tools for analyses	0		1	*Preview of tabular data*	0	
Visualizing data on maps	1	*Visualization of geo metadata*	1	*Visualization of geo metadata*	1	*Visualization of geo metadata*
Detailed reuse information	0		0		1	*Indicates purpose and user*
Quality Assurance						
Information about data quality	2	*Comprehensive quality evaluation*	0		1	*Simple quality evaluation*
Provide quality dimensions to compare datasets and its evolution	2	*Comprehensive quality evaluation*	0		0	
Interaction						
Support interaction and communication between various stakeholders	0		0		2	*Discussion platform*
Enrich data	0		0		1	*Additional community resources*
Support revisions and version history	0		1	*Metadata revision*	0	
Track reuse	0		0		2	*Linked reuse in dataset*
Performance and Architecture						
Maturity	1	*Application in a few portals*	2	*Application in many portals*	1	*Application in a few portals*
Personalization and Custom Themes	1	*Replaceable themes*	2	*Use of theme API*	1	*Replaceable themes*
Scalable Architecture	2	*Microservice architecture*	1	*Monolithic architecture*	1	*Monolithic architecture*
Score	28		21		24	

Table 1. Feature Comparison

6.3 Impact of Semantic Web Technologies

The initially required development effort was higher and partly more challenging than with more traditional approaches. Some artifacts of the Semantic Web have not yet reached the required production readiness or caught up with latest progresses in software development. This increased integration effort and required some interim solutions for providing a production system. For instance, integrating synchronous third-party libraries into our asynchronous programming model. Particularly challenging was the adoption of a triplestore as primary database. The access is implemented on a very low level via SPARQL, since a mature object-relational mapping (ORM) tool does not exist. Most of the integrity and relationship management of the data is handled on application level and needed to be implemented there, since the triplestore, unlike relational databases, cannot handle constraints directly. In addition, the SPARQL endpoint should be openly available. This currently prevents the management of closed or draft data and will require a more elaborated approach. To the best of our knowledge no (free) production triplestore is available, supporting that kind of access control on the SPARQL endpoint. Furthermore, in the Open Data domain there is no suitable and mature method to present RDF in a user interface. Hence, the transformation and processing of RDF is still required before final presentation. Usually, this presentation is domain-depended and builds on custom implementations. We solved this by applying our search service for both, strong search capabilities and immediate presentation of the data in a user front-end.
However, the overall benefits outweigh the initial barriers and efforts. With our native application of the Semantic Web data model and its definite standards via a triplestore as principal data layer, we are much more able to harness the full potential of many Open Data specifications. This particularly concerns the required implementation of DCAT-AP. The direct reuse and linking to existing vocabularies or other resources enable a more expressive and explicit description of the data, e.g. for license, policy, and provenance information. In addition, this approach increases the machine-readability. The good supply of tools for working with RDF simplifies the integration into third-party applications and creates new possibilities for browsing, processing, and understanding the data. Especially, the availability of tools for reasoning can support the creation of new insights and derived data. The native capabilities of RDF to handle multiple languages support the cross-national aspect of Open Data. The application of SHACL in connection with DQV allowed us to generate and provide comprehensive quality information in a very effective fashion. In general, the strong liaison of the Semantic Web technologies facilitates a seamless integration of the data processing pipe.

7 Conclusions and Outlook

In this paper we have presented our scalable Open Data management platform Piveau. It provides functions for Open Data publication, quality assurance, and reuse, typically conducted by public administrations, research institutes and journalists. We applied a wide range of Semantic Web technologies and principles in our solution to overcome barriers and to address functional requirements of this domain. Although the Open Data community has always leveraged specifications of the Semantic Web, our work takes a previously untaken step by designing our platform around Semantic Web technologies from scratch. This allows for a much more efficient and immediate application of existing Open Data specifications. Hence, Piveau closes a gap between formal specifications and their utilization in production. We combined this with a new scalable architecture and an efficient development lice-cycle approach. Our orchestration approach enables a sustainable and flexible creation of Open Data platforms. Furthermore, it fosters the reuse of individual aspects of Piveau beyond the scope of Open Data. We have shown that our work can compete with existing Open Data solutions and exceed their features in several aspects. We have improved the generation and provision of quality information, enhanced the expressiveness of the metadata model and the support for multilingualism. As the core technology of the European Data Portal, Piveau promotes the Semantic Web as a highly relevant concept for Europe's data economy and has proven to be ready for production and reached a high degree of maturity. Finally, our work is a relevant contribution to the 5-star deployment scheme of Open Data, which supports the concept of Linked Open Data [8]. The source code of Piveau can be found on GitLab.[17]

In the next steps, Piveau will be extended with additional features. This includes support for user interaction, data enrichment, and data analysis. The support for further Semantic Web features is also planned, e.g. compliance with the LDP specifications and the extension beyond metadata to manage actual data as RDF. Open research questions are the implementation of revision and access control on triplestore level, which cannot be satisfied yet on production-grade. In general, we aim to increase the overall readiness, broaden the target group beyond the Open Data community, and strengthen the meaning of Semantic Web technologies as core elements of data ecosystems.

Acknowledgments

This work has been partially supported by the Federal Ministry of Education and Research of Germany (BMBF) under grant no. 16DII111 (”Deutsches Internet-Institut”) and by the EU Horizon 2020 project ”Reflow” under grant agreement no. 820937. The implementation and provision of the European Data Portal is funded by the European Commission under contracts DG CONNECT SMART 2014/1072 and SMART 2017/1123.

[17] https://gitlab.com/piveau

References

1. Data Catalog Vocabulary (DCAT), https://www.w3.org/TR/vocab-dcat/
2. W3C Semantic Web Activity Homepage. https://www.w3.org/2001/sw/
3. The World of Open Data: Concepts, Methods, Tools and Experiences. Springer Science+Business Media, New York, NY (2018)
4. Albertoni, R., Isaac, A.: Data on the web best practices: Data quality vocabulary. https://www.w3.org/TR/vocab-dqv/, (Accessed 3.12.2019)
5. Amazon Web Services Inc.: Aws data pipeline. https://aws.amazon.com/datapipeline/, (Accessed 3.12.2019)
6. Archer, P., Goedertier, S., Loutas, N.: D7.1.3 - Study on persistent URIs, with identification of best practices and recommendations on the topic for the MSs and the EC (Dec 2012), https://joinup.ec.europa.eu/sites/default/files/document/2013-02/D7.1.3%20-%20Study%20on%20persistent%20URIs.pdf
7. Beno, M., Figl, K., Umbrich, J., Polleres, A.: Perception of Key Barriers in Using and Publishing Open Data. JeDEM - eJournal of eDemocracy and Open Government **9**(2), 134–165 (Dec 2017). https://doi.org/10.29379/jedem.v9i2.465
8. Berners-Lee, T.: Linked Data, https://www.w3.org/DesignIssues/LinkedData.html, (Accessed: 11.03.2019)
9. Bonér, J., Farley, D., Kuhn, R., Thompson, M.: The ractive manifesto. https://www.reactivemanifesto.org/, (Accessed 5.12.2019)
10. CKAN Association: CKAN, https://ckan.org/
11. Davis, M., Phillips, A., Umaoka, Y., Falk, C.: Bcp 47 extension t - transformed content. https://tools.ietf.org/html/rfc6497, (Accessed 03.12.2019)
12. Dragan, A.: DCAT Application Profile for data portals in Europe (Nov 2018), https://joinup.ec.europa.eu/sites/default/files/distribution/access_url/2018-11/014bde52-eb3c-4060-8c3c-fcd0dfc07a8a/DCAT_AP_1.2.pdf
13. DuraSpace Wiki: Linked (Open) Data, https://wiki.duraspace.org/display/DSDOC6x/Linked+%28Open%29+Data, (Accessed: 11.03.2019)
14. European Commision: Open data — Digital Single Market, https://ec.europa.eu/digital-single-market/en/open-data, (Accessed: 11.03.2019)
15. European Commission: DCAT-AP 1.2.1. https://joinup.ec.europa.eu/solution/dcat-application-profile-data-portals-europe/distribution/dcat-ap-121-shacl, (Accessed 3.12.2019)
16. IBM: Ibm infosphere datastage. https://www.ibm.com/products/infosphere-datastage, (Accessed 3.12.2019)
17.]init[AG und SID Sachsen: DCAT-AP.de Spezifikation, https://www.dcat-ap.de/def/dcatde/1.0.1/spec/specification.pdf, (Accessed: 11.03.2019)
18. Kantara Initiative: Federated authorization for user-managed access (uma) 2.0. https://docs.kantarainitiative.org/uma/wg/oauth-uma-federated-authz-2.0-09.html, (Accessed 3.12.2019)
19. Kirstein, F., Dittwald, B., Dutkowski, S., Glikman, Y., Schimmler, S., Hauswirth, M.: Linked data in the european data portal: A comprehensive platform for applying dcat-ap. In: International Conference on Electronic Government. pp. 192–204 (2019), https://academic.microsoft.com/paper/2967218146
20. Knublauch, H., Kontokostas, D.: Shapes constraint language (shacl). https://www.w3.org/TR/shacl/, (Accessed 3.12.2019)
21. LYRASIS: Fedora - the flexible, modular, open source repository platform. https://duraspace.org/fedora/, (Accessed 22.11.2019)

22. North Concepts Inc.: Data pipeline. https://northconcepts.com/, (Accessed 3.12.2019)
23. Open Data Team: Customizable and skinnable social platform dedicated to (open)data., https://github.com/opendatateam/udata, (Accessed: 11.03.2019)
24. Open Knowledge BE: Dcat-be. linking data portals across belgium. http://dcat.be/, (Accessed 22.11.2019)
25. Open Knowledge Foundation Labs: Data pipes. https://datapipes.okfnlabs.org/, (Accessed 3.12.2019)
26. OpenDataSoft: Open Data Solution, https://www.opendatasoft.com/solutions/open-data/, (Accessed: 11.03.2019)
27. OpenLink Software: About OpenLink Virtuoso, https://virtuoso.openlinksw.com/, (Accessed: 11.03.2019)
28. Oracle: Oracle autonomous data warehouse. https://www.oracle.com/de/database/data-warehouse.html, (Accessed 3.12.2019)
29. Publications Office of the EU: Authority tables, https://publications.europa.eu/en/web/eu-vocabularies/authority-tables, (Accessed: 11.03.2019)
30. Roffia, L., Azzoni, P., Aguzzi, C., Viola, F., Antoniazzi, F., Cinotti, T.: Dynamic linked data: A sparql event processing architecture. Future Internet **10**, 36 (04 2018). https://doi.org/10.3390/fi10040036
31. SAS Institue: Sas. https://www.sas.com/, (Accessed 3.12.2019)
32. Talend: Talend open studio. https://www.talend.com/products/talend-open-studio/, (Accessed 3.12.2019)
33. The Apache Software Foundation: Apache airflow. https://airflow.apache.org/, (Accessed 3.12.2019)
34. The OpenID Foundation: Openid connect core 1.0 incorporating errata set 1. https://openid.net/specs/openid-connect-core-1_0.html, (Accessed 3.12.2019)
35. The Scriptella Project Team: Scriptella etl project. https://scriptella.org/, (Accessed 3.12.2019)
36. Vrandečić, D., Krötzsch, M.: Wikidata: A Free Collaborative Knowledgebase. Commun. ACM **57**(10), 78–85 (Sep 2014). https://doi.org/10.1145/2629489
37. W3C: Data on the web best practices. https://www.w3.org/TR/dwbp/, (Accessed 02.12.2019)
38. W3C Wiki: LDP Implementations, https://www.w3.org/wiki/LDP_Implementations, (Accessed: 11.03.2019)
39. Wilkinson, M., Dumontier, M., Aalbersberg, et al., I.: The fair guiding principles for scientific data management and stewardship. Sci Data 3 (2016). https://doi.org/10.1038/sdata.2016.18