

Wikibase as an Infrastructure for Knowledge Graphs: the EU Knowledge Graph

Dennis Diefenbach, Max de Wilde, Samantha Alipio

▶ **To cite this version:**

Dennis Diefenbach, Max de Wilde, Samantha Alipio. Wikibase as an Infrastructure for Knowledge Graphs: the EU Knowledge Graph. ISWC 2021, Oct 2021, Online, France. hal-03353225

HAL Id: hal-03353225
https://hal.science/hal-03353225

Submitted on 23 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Wikibase as an Infrastructure for Knowledge Graphs: the EU Knowledge Graph

Dennis Diefenbach[1,2][0000−0002−0046−2219], Max De Wilde[3][0000−0002−6834−0250], and Samantha Alipio[4]

[1] The QA Company, Saint-Etienne, France
[2] Université de Lyon, CNRS UMR 5516 Laboratoire Hubert Curien, France
[3] Université libre de Bruxelles, Brussels, Belgium
[4] Wikimedia Deutschland, Berlin, Germany

Abstract. Knowledge graphs are being deployed in many enterprises and institutions. An easy-to-use, well-designed infrastructure for such knowledge graphs is not obvious. After the success of Wikidata, many institutions are looking at the software infrastructure behind it, namely Wikibase.
In this paper we introduce Wikibase, describe its different software components and the tools that have emerged around it. In particular, we detail how Wikibase is used as the infrastructure behind the "EU Knowledge Graph", which is deployed at the European Commission. This graph mainly integrates projects funded by the European Union, and is used to make these projects visible to and easily accessible by citizens with no technical background.
Moreover, we explain how this deployment compares to a more classical approach to building RDF knowledge graphs, and point to other projects that are using Wikibase as an underlying infrastructure.

Keywords: Knowledge Graph · Wikibase · EU Knowledge Graph

1 Introduction

Wikibase[1] is the software that runs Wikidata[2] [12]. Wikidata evolved into a central hub on the web of data and one of the largest existing knowledge graphs, with 93 million items maintained by a community effort. Since its launch, an impressive 1.3 billion edits have been made by 20 000+ active users.[3] Today, Wikidata contains information about a wide range of topics such as people, taxons, countries, chemical compounds, astronomical objects, and more. This information is linked to other key data repositories maintained by institutions such as Eurostat, the German National Library, the BBC, and many others, using 6 000+ external identifiers.[4] The knowledge from Wikidata is used by

[1] https://wikiba.se
[2] https://www.wikidata.org
[3] https://www.wikidata.org/wiki/Wikidata:Statistics
[4] The exact number is growing every day and can be tracked at https://w.wiki/3BSZ.

search engines such as Google Search, and smart assistants including Siri, Alexa, and Google Assistant in order to provide more structured results. While one of the main success factors of Wikidata is its community of editors, the software behind it also plays an important role. It enables the numerous editors to modify a substantial data repository in a scalable, multilingual, collaborative effort.

Because of the success of Wikidata, many projects and institutions are looking into Wikibase, the software that runs Wikidata. Their objective is mainly to reuse the software to construct domain-specific knowledge graphs. Besides this success, two main factors make Wikibase attractive: 1) the fact that it is a well-maintained open source software, and 2) the fact that there is a rich ecosystem of users and tools around it. Moreover, Wikimedia Deutschland (WMDE), the maintainer of Wikibase, has made considerable investments toward optimising the use of the software outside of Wikidata or other Wikimedia projects. Since 2019, Wikibase has had a separate product roadmap[5] and its own driving strategy based on the diverse needs of the many institutions, researchers, and individuals who depend upon the software for their projects.

In this paper, we show how Wikibase can be used as the infrastructure of a knowledge graph. While Wikibase as a standalone piece of software has been available for several years, there are not many production systems using it. We detail how Wikibase is used to host the infrastructure of a knowledge graph at the European Commission called "The EU Knowledge Graph".[6] This graph contains heterogeneous data items such as countries, buildings, and projects funded by the European Union. It is used to serve multiple services such as the Kohesio website[7] that aims to make projects funded by the EU easily accessible by citizens. Several bots help to enrich the data and to keep it up-to-date. Beside the EU Knowledge Graph, we point to other relevant Wikibase deployments.

The paper is organized as follows. In Section 2, we list some notable knowledge graphs and describe how they are deployed. In Section 3, we describe Wikibase and how it can be used to set up a local knowledge graph. In Section 4, we describe the instance that we deployed at the European Commission, including how the data is ingested, what is the current content, how it is maintained using bots, what services are offered for public consumption, and finally Kohesio, the service that is mainly served by it. In Section 5, we provide a short comparison between a typical approach to deploying knowledge graphs and our approach using a Wikibase instance. In Section 6, we present other projects that are also using Wikibase. We conclude and point to future work in Section 7.

2 Related work

Knowledge graphs [6] are data structures that are well-suited to store heterogeneous information. Many enterprises and institutions create knowledge graphs.

[5] https://www.wikidata.org/wiki/Wikidata:Development_plan#Wikibase_ecosystem
[6] https://linkedopendata.eu
[7] https://kohesio.eu

Generally, one distinguishes between Open Knowledge Graphs, which are intended to share knowledge with the general public, and Enterprise Knowledge Graphs, which are used to store and model internal or restricted knowledge.

Domain-specific Open Knowledge Graphs include, for example, the SciGraph[8] [3] which aims to aggregate metadata about the publications of Springer Nature. Recently, a knowledge graph about companies has been constructed with the aim of discovering aggressive tax planning strategies[9] [8]. Europeana[10] [4], a platform that aggregates the digitised collections of more than 3 000 institutions across Europe, releases its collections as an RDF graph.[11] The common pattern behind these knowledge graphs is that they generally are constructed by: (1) defining an underlying RDF data model, (2) integrating heterogeneous information across different data sources using this model and (3) exposing it using a triplestore.

Enterprise Knowledge Graphs are deployed at Google,[12] Airbnb,[13] Amazon,[14] LinkedIn,[15] etc. Since these graphs are not public, the technology stacks used to create and maintain them are largely unknown.

The tool closest to Wikibase is Semantic MediaWiki[16] [7], which also allows the integration and editing of knowledge in a collaborative effort. The main difference is that Semantic MediaWiki is developed for visualizing and using data within the wiki itself. Wikibase, on the other hand, has been developed to collaboratively create and maintain knowledge which then can be consumed by external applications.

3 Wikibase

In this section, we describe the different technological components that make up the core of Wikibase. It is important to understand that "Wikibase" is often used to refer to different things. We use the term Wikibase for all services and software components included in the Wikibase Docker container,[17] which is generally seen as the standard way to deploy a local Wikibase instance. All components described in this section are summarized in Figure 1.

3.1 Wikibase infrastructure

The Wikibase infrastructure consists of the following software components:

[8] https://www.springernature.com/gp/researchers/scigraph
[9] http://taxgraph.informatik.uni-mannheim.de
[10] https://www.europeana.eu
[11] https://pro.europeana.eu/page/linked-open-data
[12] https://blog.google/products/search/introducing-knowledge-graph-things-not/
[13] https://medium.com/airbnb-engineering/scaling-knowledge-access-and-retrieval-at-airbnb-665b6ba21e95
[14] https://www.amazon.science/blog/building-product-graphs-automatically
[15] https://engineering.linkedin.com/blog/2016/10/building-the-linkedin-knowledge-graph
[16] https://www.semantic-mediawiki.org
[17] https://github.com/wmde/wikibase-docker

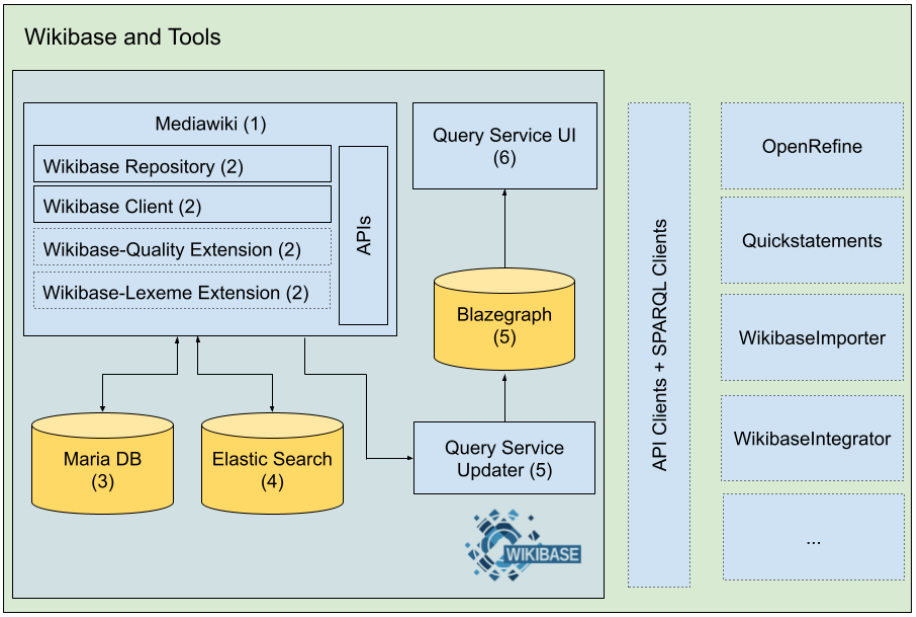


Fig. 1. Architecture of Wikibase. On the left, the core Wikibase infrastructure; on the right, the services that are constructed around it using the different MediaWiki APIs and SPARQL clients. In yellow, we have highlighted the places where the data is stored. The arrows indicate the direction of the data flows.

- MediaWiki, a wiki engine that is mainly known as the software running Wikipedia. MediaWiki started in 2002 and is continuously developed by the Wikimedia Foundation. It is mainly written in PHP and there is a vibrant ecosystem of extensions around it, i.e. components that allow the customization of how a MediaWiki installation looks and works. Today, there are more than 1 800 extensions available.
- Wikibase itself includes several of these extensions. The main extensions are the Wikibase Repository[18] and the Wikibase Client.[19] While MediaWiki was originally designed to store unstructured data, the Wikibase extensions modify it for use as a structured data repository and improve the user-friendliness of its interface. There are two more extensions that play an important role in Wikidata but which are not included in the Wikibase Docker file: the Wikibase Quality Extension[20] and the Wikibase Lexemes extension.[21] The first was designed to enable Wikibase users to define ontological constraints and detect if they are not respected at the data level. This includes, for example, the definition of domain and range constraints. The goal of the second ex-

[18] https://www.mediawiki.org/wiki/Extension:Wikibase_Repository
[19] https://www.mediawiki.org/wiki/Extension:Wikibase_Client
[20] https://www.mediawiki.org/wiki/Extension:Wikibase_Quality_Extensions
[21] https://www.mediawiki.org/wiki/Extension:WikibaseLexeme

tension is to allow the modeling of lexical entities such as words and phrases. The first extension is very useful for managing knowledge graphs and is used in the EU Knowledge Graph. The Lexemes extension, on the other hand, is not used. All these extensions are developed and maintained by WMDE and written in PHP.

- All the data is stored natively in a MariaDB[22] relational database. This includes the user management, permission management, the full log of the change history, the pages, and more.[23] The above-mentioned extensions also store data in this database, including the item list, the properties, and the changes.[24]
- The core Wikibase infrastructure includes an Elasticsearch instance. This instance is responsible for the "search-as-you-type" completion that is used to search and edit the data. Moreover, it is used for full-text search over all labels and descriptions.
- While the data is stored in a relational database, it is also exported into a triplestore which is a Blazegraph derivative maintained by the Wikimedia Foundation. Wikibase does not only provide the triplestore but tightly integrates it into the rest of the infrastructure so that changes in the relational database are directly reflected in the triplestore. A process called updater monitors the changes and reflects them in the triplestore at an interval of 10 seconds.
- The final Wikibase component is the SPARQL user interface. This offers a user interface for editing SPARQL, querying the data, and exporting the results in various formats (JSON, CSV/TSV, HTML...). It also offers different widgets for rendering the result sets in graphs, charts, maps, and more.

All of these software components make up the core infrastructure of Wikibase. The complexity is hidden inside a Docker container that is provided by WMDE, which allows for simple set up of a local Wikibase instance. While the Docker container considerably reduces the effort needed to maintain such an instance, it is still crucial to understand all components and how they interact in order to run, maintain, and customize the instance.

3.2 Tools

Around the Wikibase core infrastructure, there is a rich number of additional tools. These can be either MediaWiki extensions or external tools that are connected to MediaWiki through its APIs. In particular, there are client libraries wrapped around the Wikibase API[25] for different programming languages. The

[22] https://mariadb.com
[23] For a full database schema, one can refer to https://www.mediawiki.org/w/index.php?title=Manual:Database_layout/diagram&action=render.
[24] https://www.mediawiki.org/wiki/Wikibase/Schema
[25] https://www.mediawiki.org/wiki/Wikibase/API

most notable include Pywikibot[26] for Python, the Wikidata Toolkit[27] for Java, and the wikibase-javascript-api[28] for JavaScript.

One particular type of tools are Bots,[29] i.e. programs that edit the data in a Wikibase without human intervention. Since Wikibase is the software used by Wikidata, every tool developed for Wikidata can in theory be used also for a custom Wikibase instance, although some adaptation might be necessary. A publicly maintained list of tools around Wikidata can be found at https://www.wikidata.org/wiki/Wikidata:Tools.

Some popular tools that are used in combination with a local Wikibase instance are:

- OpenRefine[30] [11], a tool for working with messy data: cleaning it, transforming it from one format into another, and extending it with web services and external data. OpenRefine provides a native Wikidata reconciliation service that can be extended to other Wikibase instances.[31]
- WikibaseImport,[32] a Wikimedia extension for importing entities from Wikidata into a local Wikibase.
- QuickStatements,[33] a tool to import and edit data in a Wikibase using a simplified, non-programmatic language.
- WikibaseIntegrator,[34] a Python library for creating bots on top of Wikibase.
- WikibaseManifest,[35] an extension that provides an API endpoint allowing automated configuration discovery. The endpoint returns important metadata about the local Wikibase and can be used to configure external tools.
- EntitySchema,[36] an extension for storing Shape Expression (ShEx) Schemas on wiki pages. ShEx [9] is a language for validating and describing RDF.

4 The EU Knowledge Graph

As described above, it is relatively straightforward to set up an empty Wikibase instance with many services that are offered out of the box. In this section, we describe the workflow that we followed to build the EU Knowledge Graph which is available at https://linkedopendata.eu. This includes how we initialized the graph, how we ingest data, how we maintain it, which services we provide, and which services rely on it.

[26] https://www.mediawiki.org/wiki/Manual:Pywikibot
[27] https://www.mediawiki.org/wiki/Wikidata_Toolkit
[28] https://github.com/wikimedia/wikibase-javascript-api
[29] https://www.wikidata.org/wiki/Wikidata:Bots
[30] https://openrefine.org/
[31] https://github.com/wetneb/openrefine-wikibase/
[32] https://github.com/Wikidata/WikibaseImport
[33] https://www.wikidata.org/wiki/Help:QuickStatements
[34] https://github.com/LeMyst/WikibaseIntegrator
[35] https://www.mediawiki.org/wiki/Extension:WikibaseManifest
[36] https://www.mediawiki.org/wiki/Extension:EntitySchema

4.1 Creating seed entities and relations

While a Wikibase instance is usually intended to ingest knowledge that is not contained in Wikidata, the latter still generally provides entities and properties that are relevant to model domain-specific knowledge. Therefore, as a first step, we identified entities in Wikidata that are relevant for the European Commission. This includes concepts like the European Union, member states, capital cities, heads of states, European institutions, and more. We imported these entities directly into our local installation using the WikibaseSync[37] tool developed during the course of the project. The tool creates two properties in the Wikibase (external identifiers) and generates, for each item and property imported from Wikidata, a corresponding entity in the local Wikibase instance.

The two external identifiers keep track of the correspondence between the items and relations in the Wikibase and in Wikidata (in particular these can be used to translate a SPARQL query over Wikidata to one over Wikibase). This first step enables the reuse of many items and properties from Wikidata that can be further used to model domain-specific knowledge. In general, we followed the policy to always reuse items and relations from Wikidata whenever possible, i.e. we introduced new items and relations in the Wikibase only if they were not preexisting in Wikidata.

4.2 A typical data import

In this section, we describe a typical data import workflow. As an example, we detail how we imported data about the buildings that are occupied by the European Commission in Brussels. This includes the following steps:

- **Data collection**: All information about the buildings is made available through an API. A snippet is shown in Figure 2.
- **Modeling**: To model this piece of data, we need the concepts of building and office, as well as properties like address, opening hours, and occupant. Whenever possible, we take Wikidata entities/properties or reuse existing entities/properties in the Wikibase. In particular, the following concepts already exist in Wikidata and were imported into the Wikibase using the WikibaseSync tool:
  - Building (Q41176 in Wikidata; Q8636 in the Wikibase)
  - Office (Q182060 in Wikidata; Q244596 in the Wikibase)
  - the property occupant (P466 in Wikidata; P641 in the Wikibase)
  - and some more...

  Note that by ingesting Wikidata knowledge, we know for example that a "building" is called "Gebäude" in German, that "office" is a subclass of a "workplace", and that "phone number" is also expressed as "telephone number". In particular, this means that part of the knowledge is created by people outside the European Commission. In more specific domains, a close interaction with domain experts might be necessary to correctly understand and model the data.

[37] https://github.com/the-qa-company/WikibaseSync

```
{
   "type": "OFFICES",
   "code": "BU25",
   "name": "Beaulieu 25",
   "photoLink": "BU25.jpg",
   "occupants": "CNECT",
   "contactList": {
      "contacts": [
         {
            "name": "Réception",
            "phone": "+32229 53818"
         }
      ]
   },
   "buildingAddress": {
      "streetAddress": "Avenue de Beaulieu 25",
      "gpsCoordinates": {
         "latitude": 50.814347,
         "longitude": 4.412298
      }
   }
}
```


Fig. 2. Snippet of the JSON API response describing buildings occupied by the European Commission.

- **Attention to identifiers**: When importing data, we make sure to always insert external identifiers so we can easily link the newly ingested data to the original data repository. For the particular building in Figure 2 for example, we want to use the code "BU25" as an external identifier.
- **Linking**: Some entities are already present in the Wikibase, such as the occupant "CNECT" which is the Directorate-General (DG) of the European Commission for Communications Networks, Content, and Technology. The most common strategy that we use is a simple key linking, using external identifiers provided in Wikidata and/or the Wikibase instance. We take advantage of the fact that Wikidata has become established as a reference point for many datasets. The more than 6 000 external identifiers make this possible.
- **Import**: Finally, we import the data itself based on the chosen data model. The import is performed with Pywikibot.[38] The entity corresponding to the initial JSON snippet[39] is shown in Figure 3.

[38] https://www.mediawiki.org/wiki/Manual:Pywikibot
[39] https://linkedopendata.eu/entity/Q242372

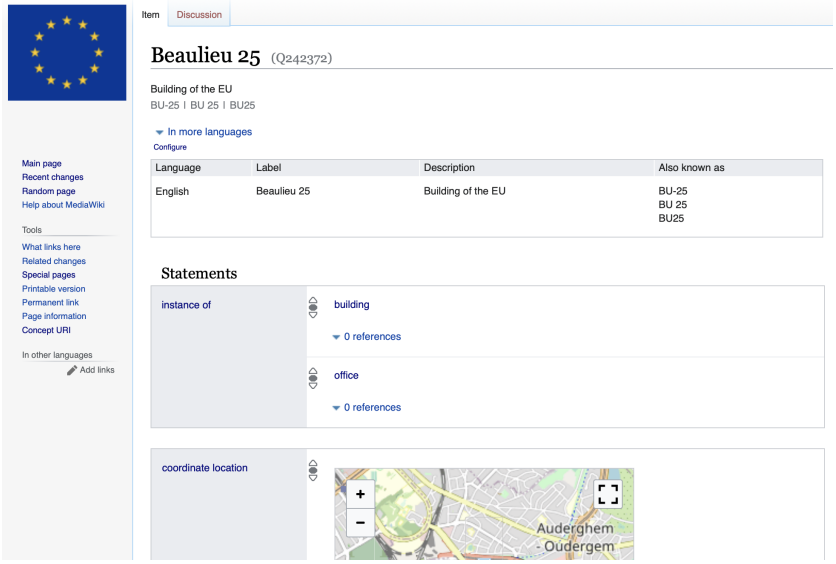


Fig. 3. View of the BU25 building in the EU Knowledge Graph.

4.3 Current content

The current content of the EU Knowledge Graph has been imported as described above. It includes:

- institutions of the European Union (like the European Parliament and the Council of the European Union);
- countries of the world, and in particular member states of the European Union (like Hungary and Italy);
- capital cities of European countries (like Athens and Tallinn);
- DGs of the European Commission (like DG CNECT and DG REGIO);
- buildings, canteens, cafeterias and car parks of the European Commission;
- the largest part of the graph is composed of 705 108 projects[40] and 112 688 beneficiaries[41] of projects funded by the European Union under Cohesion Policy. This data has been aggregated from more than 40 Excel and CSV sheets provided in a non-standardized format by the member states of the European Union. These files describe the projects funded on a national or regional level, following EU regulations.

To date, the whole dataset comprises 96 million triples and 1 845 properties.

[40] Like https://linkedopendata.eu/entity/Q77409
[41] Like https://linkedopendata.eu/entity/Q2529763

4.4 Bots: enriching and maintaining the data

Important steps in constructing a knowledge graph include data enrichment as well as maintaining data freshness. Following Wikidata practices, we deployed a number of bots which work independently and each focus on a specific task. These bots include:

- **Wikidata Updater Bot**: As explained in Section 4.1, some of the entities and relations of the EU Knowledge Graph come from Wikidata. The Wikidata Updater Bot makes sure that changes in Wikidata are transferred automatically to the EU Knowledge Graph. If an edit is made to an entity in Wikidata that is also available in the EU Knowledge Graph, this edit is directly transferred with a delay of 5 minutes. This means that part of the knowledge is maintained by the Wikidata Community, e.g. for new heads of state or heads of government of a country. There are currently almost 130 000 entities and more than 1 800 properties maintained by Wikidata editors.
- **Merger Bot**: It can happen that a Wikidata entity is linked twice. The Merger Bot takes care of merging the two entities and redirecting one to the other.
- **Translator Bot**: An important element for the European Commission is that the knowledge can be made available in multiple languages. The Translator Bot translates specific entities from one language to another. This bot relies on machine translation provided by the eTransation tool.[42]
- **Geocoding Bot**: This bot is responsible for inferring geographic coordinates from the postal code. For example, if a project includes only a postal code, the corresponding geographic coordinates are inferred with Nominatim.[43]
- **Beneficiary Linker Bot**: One key piece of information for funded projects is the beneficiary. However, this often consists of a simple string. The objective of the Linker Bot is to detect if the beneficiary is also available as an entity in Wikidata and attempt to link it. The project https://linkedopendata.eu/entity/Q77409 for instance indicates as a beneficiary "PARAFIA ŚW. ŁUKASZA EWANGELISTY W LIPNICY WIELKIEJ". The Linker Bot, based on machine learning, identifies the following Wikidata entity as a match: https://www.wikidata.org/entity/Q11811090. This allows us to enrich the data and to provide links to external sources. In this case, we are able to infer that the beneficiary is a parish of the Roman Catholic Archdiocese of Kraków whose website is http://www.parafia-lipnicawielka.pl/.
- **Beneficiary Classifier Bot**: This bot is responsible for classifying beneficiaries into public and private entities. This information is important for decision makers and for understanding how the money is spent.
- **NUTS Bot**: This bot is responsible for inferring the NUTS3[44] statistical region in which a project is contained from its geographic coordinates.

[42] https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eTranslation
[43] https://nominatim.openstreetmap.org
[44] https://ec.europa.eu/eurostat/web/nuts/background

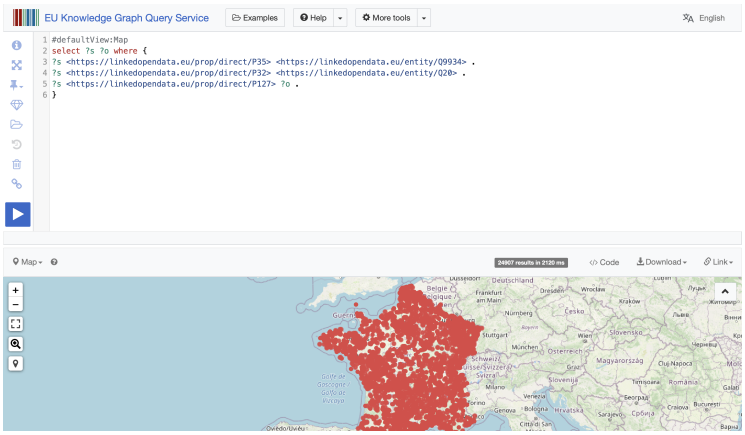


Fig. 4. Query displaying all projects in France funded by the EU under Cohesion Policy, using the query service available at https://query.linkedopendata.eu.

4.5 Services

While it is important to collect and maintain the knowledge, it is also crucial to make it easily consumable. Besides the user-friendly interface of Wikibase, we offer three ways to consume the data:

1. Data Exports: we provide full dumps of the data after the Wikidata fashion. The dumps are available at https://data.linkedopendata.eu. Moreover, we provide CSV/Excel exports of specific parts of the data for people not familiar with RDF.
2. Query Service: the standard query service of the Wikibase is available at https://query.linkedopendata.eu. Like Wikidata, it allows for the retrieval and visualisation of information. A screenshot is displayed in Figure 4.
3. Question Answering Service: we offer QAnswer [2] as a question answering service. It enables access to data in the knowledge graph via natural language queries. The service is available at https://qa.linkedopendata.eu and is shown in Figure 5.

4.6 Kohesio

Currently, the EU Knowledge Graph is mainly used as the data repository of the Kohesio project available at https://kohesio.eu (see Figure 6). Kohesio aims to collect the data of projects funded in the frame of the EU Cohesion Policy, which supports tens of thousands of projects across Europe annually. This is done through funding programmes whose management is shared between national and regional authorities on the one hand, and the European Commission on the other hand. Kohesio is still under development and is scheduled to be launched officially during the first quarter of 2022.

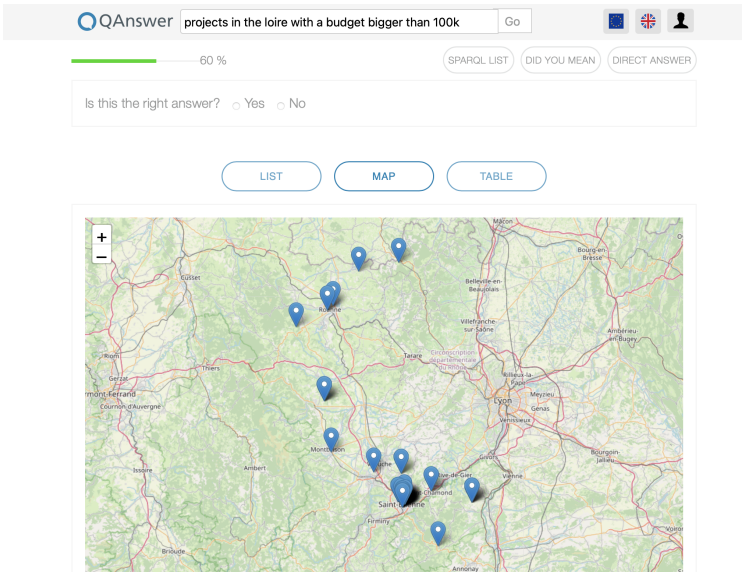


Fig. 5. Question answering service https://qa.linkedopendata.eu for the question "projects in the Loire department with a budget higher than 100 000 euros".

The projects are imported from several files published by the member states, aligned into a common data model, and enriched with additional information using bots. All data used by the website is extracted from the EU Knowledge Graph via SPARQL queries, exposed as REST APIs.[45]

5 Comparing classical approach vs Wikibase

In this section, we compare at a high level the differences between a classical RDF knowledge graph deployment and a Wikibase deployment. By a classical RDF deployment, we mean a knowledge graph that is constructed and maintained as described in Section 2: (1) defining an underlying RDF data model, (2) integrating heterogeneous information across different data sources using the model, and (3) exposing it using a triplestore. We summarize the differences in Table 1.

In a classical deployment, Semantic Web technologies are central. This allows, for example, the usage of reasoning and RDF-related technologies like SHACL, which is not the case for Wikibase as it does not natively store the data as RDF but only exposes it in this format. Despite this difference, the Wikibase community runs into problems that are similar to those encountered in the Semantic Web community. They address some of these issues in a different way, such as the Wikibase Quality Extensions mentioned above.

[45] Most of the code is published as open source at https://github.com/ec-doris

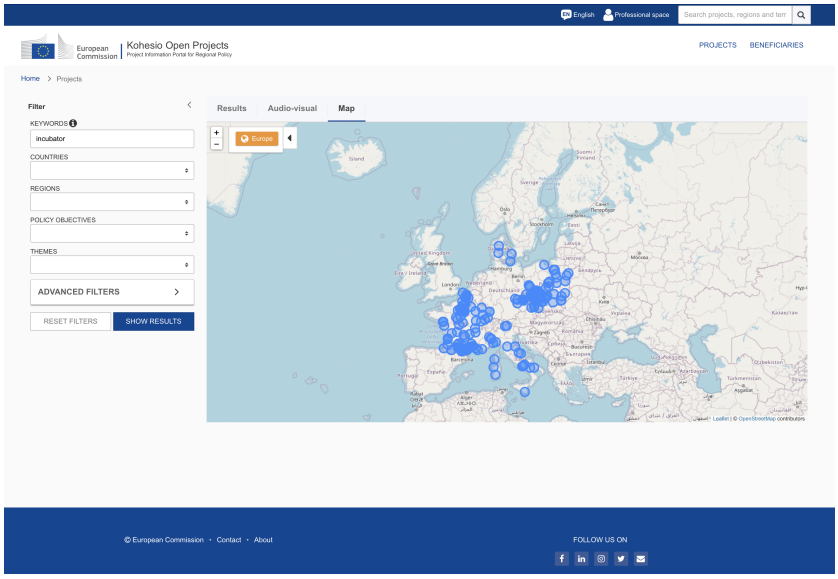


Fig. 6. Kohesio interface showing projects around Europe about "incubators".

One of the drawbacks of Wikibase is that it does not allow the reuse of external RDF vocabularies. Entities and properties will always be QIDs and PIDs. The only available workaround is to create items and properties in Wikibase, and indicate using a custom property that they are equivalent to some RDF vocabulary. Moreover, the data model of Wikibase is more restrictive than the general RDF data model. For example, blank nodes are not allowed and the datatype of the object of a property must be defined at creation time. It is therefore not possible to define a property that has both URIs and literals as objects. The data model is also more rigid because it restricts to one specific model for reification, namely n-ary [5]. This implies that it is not possible to easily import an RDF dataset into Wikibase. Additionally, a limitation of Wikibase is that the SPARQL endpoint is read-only, making it impossible to insert or update data via SPARQL. All data has to be ingested using the APIs of Wikibase.

The main advantage of Wikibase is that it offers a series of services out of the box. These include: built-in visualizations in the SPARQL endpoint (like in Figure 4), an automatic full-text search as well as a search-as-you-type functionality over the items and properties, and a simple interface to edit the data even by non-expert users. All these functionalities can also be offered in classical RDF deployments, but need special infrastructure. Another important feature is full tracking of changes. It is therefore always possible to see who contributed to the knowledge. This is important in scenarios were multiple people edit the data. We are not aware of a solution that achieves this functionality in a classical deployment.

In general, one can say that a classical deployment allows full flexibility but requires a lot of specific infrastructure to provide functionalities like data visualisation, editing, etc. Conversely, Wikibase is more rigid but provides many out-of-the-box services, as well as a deep integration into a well-established ecosystem.

	Classical approach	**Wikibase**
RDF support	Full support	The information can be dumped as RDF but is not natively in this format. RDF data is hard to ingest.
Reuse of external vocabularies and ontologies	Full support	One cannot use external vocabularies directly, but only align them with new properties in the Wikibase.
Scalability	Depending on the underlying triplestore	Ingesting large datasets is time-consuming but some projects are trying to address this issue[46] [10].
Updating queries	Relying on SPARQL	Not possible over SPARQL, since the endpoint is read-only. Only through the Wikibase APIs.
Data model	Flexible	Rigid. For example, the reification model is fixed but well established.
Visualisation	A particular software has to be installed[47]	Out-of-the-box
Editing the data	A plugin is needed	Out-of-the-box
Search	A plugin is needed	Out-of-the-box with Elasticsearch
Track changes	Unclear	Out-of-the-box
Recent changes	Unclear	Out-of-the-box

Table 1: Classical RDF infrastructure vs Wikibase

6 In-use Wikibase instances

Besides the EU Knowledge Graph, there are several Wikibase instances used by different institutions and communities. These deployments are either in a testing, pilot, or production phase. On a voluntary basis, it is possible to register a Wikibase instance in the Wikibase Registry.[48] Here are some notable projects:

– L'Agence bibliographique de l'enseignement supérieur (ABES) and the Bibliothèque nationale de France (BnF) are currently building a shared platform for collaboratively creating and maintaining reference data about entities (persons, corporate bodies, places, concepts, creative works, etc.) which will

[46] Like https://github.com/UB-Mannheim/RaiseWikibase.
[47] Such as the Virtuoso Faceted Browser.
[48] https://wikibase-registry.wmflabs.org (the full list of instances can be accessed at https://tinyurl.com/y8xun3wy).

be initially used by the Bibliothèque nationale de France and all the French Higher Education Libraries, and in a second phase by other cultural institutions (archives, museums). Its initial deployment is planned for 2023.

- The Enslaved.org project [13], available at https://enslaved.org/, aims to track the movements and details of people in the historical slave trade. The main objective of the project is to allow students, researchers, and the general public to search over numerous databases in order to reconstruct the lives of individuals who were part of the slave trade. The Enslaved.org project uses Wikibase as the main infrastructure to integrate the data from heterogeneous sources. The instance is available at https://lod.enslaved.org. QuickStatements is mainly used to clean, ingest, and model the data. The project was released in December, 2020.
- The Deutsche Nationalbibliothek[49] (DNB) is running a multi-year pilot to provide their Integrated Authority File (GND) with an alternative website. The objective is to make the free structured authority data easier to access and interoperable, and Wikibase is seen as a user-friendly solution to host and maintain the GND containing more than 9 million items. QuickStatements and Wikidata Integrator have been tested to ingest the data into the Wikibase, although a quicker custom application is now being developed. Besides the instance containing the actual authority files, a second Wikibase instance will provide all rules and regulations as structured data items and properties. This will help improve the usability of the documentation and the quality of the data processed in the first instance via shared schemas. The pilot started in 2019 and is planned to go live in 2023.
- The Archives of Luxembourg are experimenting with Wikibase to integrate data from 8 different GLAM[50] institutions (like the Archives Nationales de Luxembourg and the Bibliothèque nationale de Luxembourg). These institutions will publish their catalogue in the CIDOC Conceptual Reference Model (CRM), an extensible ontology for concepts and information in the cultural heritage domain. This data can then be ingested into a Wikibase and synchronised across institutions.
- Factgrid[51] is run by the Gotha Research Centre Germany at the University of Erfurt. It started out as a project to track the activities of the Illuminati, but it is now a collaborative, multilingual digital humanities project that collects historical research data. It uses Wikibase and has over 150 active community members.
- Wikimedia Commons uses Wikibase to enhance over 57 million CC0 media files with structured data.[52] Wikibase users can easily link entities within their instance to relevant images on Commons. A SPARQL endpoint is also provided.[53]

[49] https://www.dnb.de
[50] Galleries, libraries, archives, and museums.
[51] https://database.factgrid.de/wiki/Main_Page
[52] https://commons.wikimedia.org/wiki/Commons:Structured_data
[53] https://wcqs-beta.wmflabs.org

- Rhizome[54] is dedicated to the preservation and promotion of digital art and was among the earliest adopters of Wikibase in 2015. Wikibase's flexible data model is used to describe a unique catalog of internet artworks with specialized preservation metadata.
- The Centre for Historical Research and Documentation on War and Contemporary Society (CegeSoma[55]) in Belgium launched a pilot Wikibase instance in the context of the ADOCHS project[56] to evaluate its added value for the management of names authority files. In the context of her PhD [1], Chardonnens explored several options and documented all configuration choices on her blog Linking the Past.[57] Based on this successful experiment, a new project about members of the Resistance is about to start.

7 Conclusion and future work

In this paper, we have presented how Wikibase can be used as an infrastructure for knowledge graphs. We have shown that while Wikibase is not as flexible as a traditional RDF deployment, it offers many out-of-the-box services that are either necessary or convenient for deploying a knowledge graph infrastructure. One of the biggest advantages is that it allows non-expert users to directly access the knowledge graph. Moreover, it deeply integrates into an ecosystem of tools and libraries that are widely used (mainly for Wikidata).

Wikibase development in the near-term will be focused around two core areas. The first is improving the installation, setup, and maintenance experience for Wikibase administrators. This includes – but is not limited to – establishing a regular, predictable release cycle; improving documentation around software installation and updating; creating an improved deployment pipeline for the software; and publishing improved Docker images. The second development focus for Wikibase is around the concept of federation. In the Wikibase context, federation refers to enabling different Wikibases to link their content (e.g. entities), query across instances, or share ontologies. Most often, Wikibase projects express a desire to enhance their local instance by linking with the vast amount of general-purpose knowledge on Wikidata. For this reason, WMDE will continue its earlier work[58] by making it possible to access and reuse Wikidata's properties in combination with a local data model. These efforts will lay the groundwork for more robust sharing and linking of data between Wikibases.

[54] https://rhizome.org/art/artbase/
[55] https://www.cegesoma.be
[56] https://adochs.arch.be
[57] https://linkingthepast.org
[58] https://doc.wikimedia.org/Wikibase/master/php/md_docs_components_repo-federated-properties.html

Acknowledgements We would like to thank: Anne Thollard who initiated the Kohesio project and drove the use case internally at the European Commission; the team at DG REGIO for their support as domain experts and their precious advice; the team at DG CNECT for their initiative in investigating such an innovative approach and the technical deployments of the Kohesio user interface; DIGIT for following the project and giving feedback, in particular the helpful discussions with Ivo Velitchkov; Georgina Burnett and Jens Ohlig who coordinated the relationship between Wikimedia Deutschland and the European Commission. Finally we would like to thank the whole team of WMDE for providing great open source software, as well as the Wikidata editors for all the knowledge that they are providing every day!

References

1. Chardonnens, A.: La gestion des données d'autorité archivistiques dans le cadre du Web de données. Ph.D. thesis, Université libre de Bruxelles (2020)
2. Diefenbach, D., Both, A., Singh, K., Maret, P.: Towards a question answering system over the semantic web. Semantic Web **11**(3), 421–439 (2020)
3. Hammond, T., Pasin, M., Theodoridis, E.: Data integration and disintegration: Managing springer nature scigraph with SHACL and OWL. In: ISWC (2017)
4. Haslhofer, B., Isaac, A.: data. europeana. eu: The europeana linked open data pilot. In: International Conference on Dublin Core and Metadata Applications. pp. 94–104 (2011)
5. Hernández, D., Hogan, A., Krötzsch, M.: Reifying RDF: What works well with wikidata? SSWS@ ISWC **1457**, 32–47 (2015)
6. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutierrez, C., Gayo, J.E.L., Kirrane, S., Neumaier, S., Polleres, A., Navigli, R., Ngomo, A.C.N., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., Zimmermann, A.: Knowledge Graphs (2021)
7. Krötzsch, M., Vrandečić, D., Völkel, M.: Semantic mediawiki. In: International semantic web conference. pp. 935–942. Springer (2006)
8. Lüdemann, N., Shiba, A., Thymianis, N., Heist, N., Ludwig, C., Paulheim, H.: A Knowledge Graph for Assessing Agressive Tax Planning Strategies. In: ISWC. pp. 395–410. Springer (2020)
9. Prud'hommeaux, E., Labra Gayo, J.E., Solbrig, H.: Shape expressions: an RDF validation and transformation language. In: ISWC. pp. 32–40 (2014)
10. Shigapov, R., Mechnich, J., Schumm, I.: Raisewikibase: Fast inserts into the BERD instance (2021)
11. Verborgh, R., De Wilde, M.: Using OpenRefine. Packt Publishing Ltd (2013)
12. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)
13. Zhou, L., Shimizu, C., Hitzler, P., Sheill, A.M., Estrecha, S.G., Foley, C., Tarr, D., Rehberger, D.: The enslaved dataset: A real-world complex ontology alignment benchmark using wikibase. In: CIKM. pp. 3197–3204 (2020)