Sustainability of (Open) Data Portal Infrastructures

A Distributed Version Control Approach to Creating Portals for Reuse

This study has been prepared by the University of Southampton as part of the European Data Portal. The European Data Portal is an initiative of the European Commission, implemented with the support of a consortium led by Capgemini Invent, including Intrasoft International, Fraunhofer Fokus, con.terra, Sogeti, 52North, Time.Lex, the Lisbon Council, and the University of Southampton. The Publications Office of the European Union is responsible for contract management of the European Data Portal.

For more information about this paper, please contact:

European Commission
Directorate General for Communications Networks, Content and Technology
Unit G.1 Data Policy and Innovation
Daniele Rizzi – Policy Officer
Email: daniele.rizzi@ec.europa.eu

European Data Portal
Gianfranco Cecconi, European Data Portal Lead
Email: gianfranco.cecconi@capgemini.com

Esther Huyer
Email: esther.huyer@capgemini.com

Written and reviewed by:
Luis-Daniel Ibáñez
Email: l.d.ibanez@soton.ac.uk

Johanna Walker
Email: j.c.walker@soton.ac.uk

Mike Hoffman
Email: m.hoffman@soton.ac.uk

Elena Simperl
Email: elena.simperl@kcl.ac.uk

Last update: 02.03.2020
www: https://europeandataportal.eu/
@: info@europeandataportal.eu

DISCLAIMER
By the European Commission, Directorate-General of Communications Networks, Content and Technology. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the Commission. The Commission does not guarantee the accuracy of the data included in this study. Neither the Commission nor any person acting on the Commission's behalf may be held responsible for the use, which may be made of the information contained therein.

Luxembourg: Publications Office of the European Union, 2020

OA-02-20-165-EN-N ISBN: 978-92-78-42148-9 doi: 10.2830/151838

Note: this document is part of a series of research reports developed on the topic of “Sustainability of (open) data portal infrastructures”, all of which are available on the European Data Portal at https://www.europeandataportal.eu/en/impact-studies/studies .

The series is made of the following reports:

1. A summary overview
2. Measuring use and impact of portals
3. Developing Microeconomic Indicators Through Open Data Reuse
4. Automated assessment of indicators and metrics
5. Assessment of Funding Options for Open Data Portal Infrastructures
6. Open data portal assessment using user-oriented metrics
7. Leveraging distributed version control systems to create alternative portals

Abstract

In the second half of 2019, 42% of the European Data Portal visitors reached it through a commercial search engine. This entails two side effects that need to be carefully considered. First, competition for traffic arises among sources, portals and meta-portals. Secondly, if users rely on big providers' dataset search to find datasets, then the discovery dimension of portals is at risk of becoming obsolete.

To counteract these effects, we argue that portals need to move forward in two directions: (1) satisfy information needs beyond "find a specific dataset" and (2) strengthen the co-location dimension of portals. This report considers the second part: co-location of tools and other useful capabilities and resources that empower re-users to develop and share work and improvements on datasets, that in turn can be used to increase measurability and search capabilities.

This report describes the potential of “community dataspaces” as environments to foster collaboration from re-users around datasets that add value to portals, and demonstrates a prototype based on a on top of a Distributed Version Control (DVC) system.

Table of Contents

1. Introduction

Previous studies on the future of open data portals, including the work above, suggest that co-location of tools and promotion of data reuse are two of the aspects where current portals struggle the most. Many portals, including harvesters such as the EDP, cover large numbers of datasets which are heterogeneous in terms of size, format, quality and publication environment. In addition, publishers do or cannot make any assumptions about the scenarios in which the data will be used and tailor their processes and technologies accordingly. This leads to a trade-off. In theory, by not privileging some scenarios over others, publishers maximise use; in practice, this means that releasing the data follows a one-size-fits-all approach, which often creates substantial overheads down the data value chain when data has to be tediously transformed and curated for particular applications or skill sets.

The current flow of datasets and metadata in the open data ecosystem directly includes only source organisations, portals, meta-portals and commercial search engines, leaving re-users out (Figure 1). This is a missed opportunity for (1) collecting re-use data for datasets beyond downloads that would be valuable input for the metrics developed in the accompanying report Ensuring Sustainability of Open Data Portals: Automated Assessments for Metrics and Indicators and (2) take advantage of re-users' skills and work to crowdsource quality improvements and useful transformations, decentralising part of the responsibility from source organisations, that may not have the time, resources or knowledge to understand and execute actions towards improving the re-usability of datasets. Often, different re-users independently develop and apply similar transformations to the same data, whereas a sharing and contribution infrastructure would have saved time and effort.

Furthermore, the emergence of dataset-specific search capabilities of established Web search engines, like the recently launched Google Dataset Search changes the board for open data portals. Most users are very much accustomed to do most of their searches through their favourite commercial provider and looking for datasets or related content is not the exception. In the second half of 2019, 42% of the European Data Portal visitors reached it through a commercial search engine. This entails two side effects that need to be carefully considered. First, competition for traffic arises among sources, portals and meta-portals. Look for example at the Google Dataset Search page of the "Datos Demográficos Totales de Zaragoza" dataset on Figure 2. Google has detected the same dataset indexed by four different organisations: the original source (zaragoza.es), the national portal (datos.gob.es), and two meta-portals, the European Data Portal and the Open Data Portal Watch (data.wu.ac.at). A particular order is not currently enforced, but it seems natural that as pages on portals point at the original source, the latter would be ranked first by a PageRank based algorithm. Alternatively, when the original source is less known and there are more links to portals from other pages on the Web, the portal with the best search engine optimisation would get the prime spot, and with it, most of the traffic. The ultimate decision on ordering corresponds to the owner of the search engine and there is little public sector decision makers can do about it. But even if they could have a say, on what grounds one should favour one portal over the other?

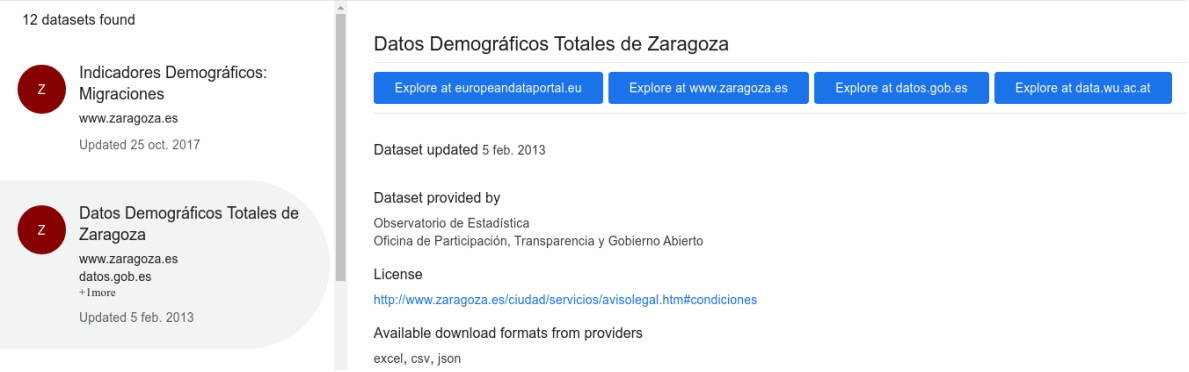

Figure 1: Google Dataset Search page of "Datos Demográficos Totales de Zaragoza". It includes links to the original source and three portals that indexed it.

The second side-effect follows straightforward from the first one: if users rely on big providers' dataset search to find datasets, then the discovery dimension of portals is at risk of becoming obsolete. In other words, if the only added value that portals and meta-portals provide is to answer single dataset queries in the style of "Demographic data of Zaragoza", they are no match for established search engines and will eventually become irrelevant.

To counteract these effects, we argue that portals need to move forward in two directions: (1) satisfy information needs beyond "find a specific dataset" and (2) strength the co-location dimension of portals. The first direction can be developed by taking advantage of the fact that portals have a general view of the datasets they harvest from their respective geographic areas to strength the “Link Data” dimension: links, relationships and similarity scores among datasets could be used to power more advanced search and discovery interfaces that create a differentiator with respect to commercial dataset search engines, an approach that is already being experimented by the European Data Portal. However, automated linking from the higher-level perspective of portals eventually reaches a cap: many relationships and links come to light only after a human re-user has put them together in response to a concrete use case. This leads to the second direction: co-location of tools and other useful capabilities and resources that empower re-users to develop and share work and improvements on datasets, that in turn can be used to increase measurability and search capabilities.

In the following, we will describe our vision of “community dataspaces” as environments to foster collaboration from re-users around datasets that add value to portals, and how we prototyped one on top of a Distributed Version Control (DVC) based system, as the ones commonly used by Open Source Software communities.

2. Community data spaces

The core of our vision is the concept of ‘community dataspaces’, that we will often abbreviate as CDSs. Community dataspaces are virtual environments that co-locate technical and social tools that can be used to create communities around single or related datasets, share or co-develop derived datasets and source code of re-uses and establish links with other datasets or other communities. Communities may be created around datasets from a specific domain, or bring together datasets from disparate sources and

categories to implement one or more re-use cases, or with the goal of finding links and similarities among them. These spaces could be hosted by community members themselves (as open source packages distributed by portals) or by portals and meta-portals. In the latter case, users should register in the portal to be able to create/join a dataspace, then, dataspace owner(s) add datasets to the dataspace through links available on the portal or upload them directly. The community can then use tools for interlinking or processing datasets (ideally available from the portal itself) or develop their own in the same space. We argue that community data spaces can bring the following benefits: (i) Increase the quality of datasets published in portals. Communities that put datasets to use are in a better position to discover and fix errors, and to find links between datasets maintained by different organisations; (ii) Results of a community's work, processes and discussions are registered on the dataspace for the benefit of the portal and its users, potentially creating a network effect, attracting more users to engage with communities and datasets; (iii) Provide data owners with a set of more advanced tools to handle their interaction with re-users and facilitate the integration (or rejection) of changes and derived datasets suggested by communities.

Figure 2 shows how we envision the new flow of data when CDSs join the ecosystem. PSI organisations that own data publish datasets in their own servers together with the necessary metadata to remain discoverable by other parties, just as they currently do. Datasets are added to one or more community dataspaces[1], hosted or indexed by meta-portals. Commercial dataset search engines would index both the original sources and the community spaces, but now that there is a clear difference between what the source organisation provides (the dataset and a proof that the original version was produced by them) and what the community space offers (co-development and derived datasets). We expect that it will be in the interest of commercial dataset search engines to make this difference. This could be further facilitated by updating DCAT and schema.org vocabularies to explicitly define community dataspaces as entities of their own right.

[1] Note however that we don't mean a portal should *only* host community data spaces. They should continue harvesting datasets towards their own search and interlinking activities.

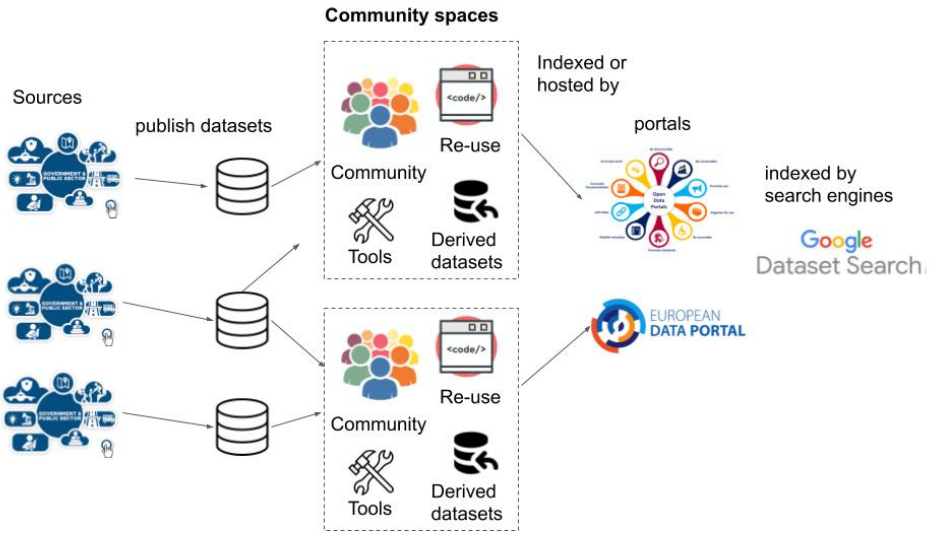

Figure 2: Users join spaces organised around datasets and share tools, develop services and apps, and derive further datasets

Not every collection of datasets will be amenable to for a community of practice around it. Communities are about people with shared goals or interests - in the EDP case, the datasets are all government datasets and the community is most likely to be formed of users of government data or open data tech developers. In other cases, for instance Kaggle, the community consists of machine learning enthusiasts. They search and share datasets in any domain alongside machine learning tools to process them, and develop 'kernels', code notebooks that describe machine learning processing based on these datasets, as shown in Figure 3. In the context of open government data, code notebooks with a strong emphasis on visualization and interlinking could be used instead.

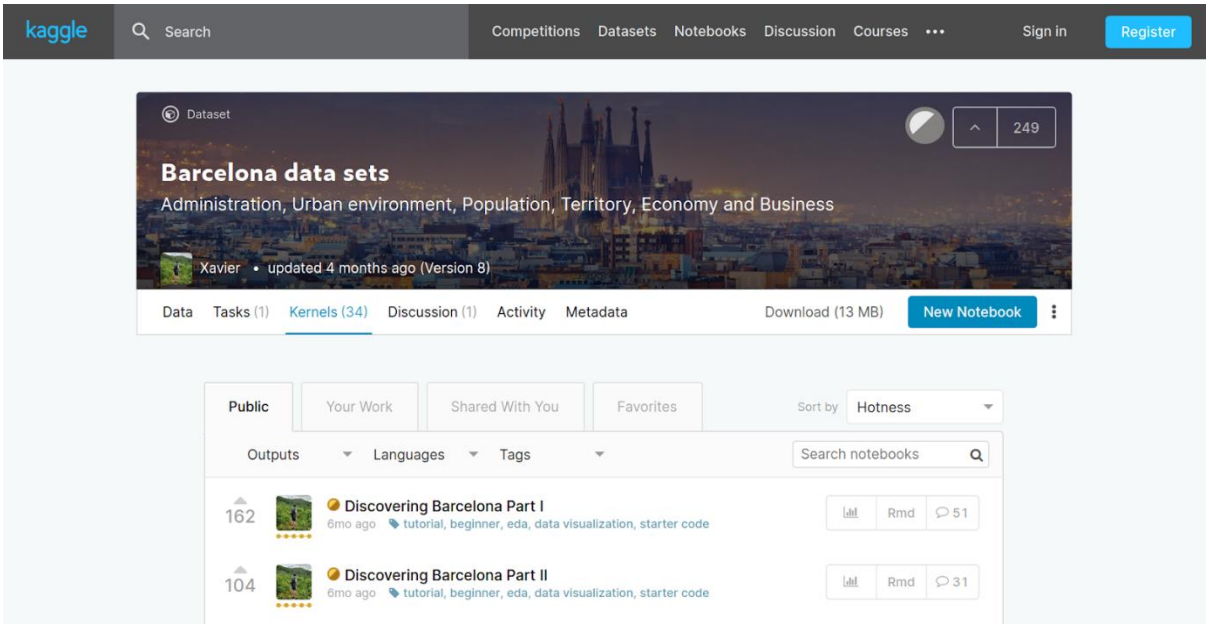

Figure 3: Kaggle page of "Barcelona data sets" associated 'kernels', machine learning/data science code notebooks contributed by members of the community

3. Extending Distributed Version Control DVC) systems to implement community data spaces

Systems based on Distributed Version Control like Github and Gitlab have been used by open source software practitioners to host their projects and foster collaboration from users of the software around the world. Besides technical version control capabilities, these systems also include other social tools useful for developers: issue trackers, wikis, forums, pull requests, project and release management, etc, with their positive impact on projects documented by industry and academia[2]. We think the most natural way to implement community data spaces is to extend current DVC systems, for the following reasons:

- Processing and re-using data often requires code, DVC systems out-of-the-box capabilities enable sharing and developing code around datasets for free. Better provenance tracking is also possible[3]
- Social tools like forums, wikis, issue trackers, etc, are straightforwardly applicable to the case of datasets
- Version control and track change capabilities used in code help data owners to manage releases, and to assess what community contributions can be integrated into the original datasets
- DVC systems are popular in the software development and data science communities, reducing the friction of moving to a different paradigm to collaborate around datasets.

[2] https://octoverse.github.com/

[3] https://blog.okfn.org/2013/07/02/git-and-github-for-data/

However, DVC systems were designed for code, and not datasets. Therefore, several extensions need to be implemented to facilitate data discovery, sensemaking and interlinking:

- Be discoverable: Search engines in VC environments are optimized for source code and would not be effective for datasets. DVC requires enhancement as follows:
    - Define conventions for the types of quality problems that are common in datasets and configure the version control technology to use them.
    - Automatic generation of the DCAT data catalogue of the portal, so it can be served from a SPARQL endpoint or consumed by a metaportal or search engine.
- Publish metadata: To encourage the improvement of metadata by the community, automatically generate issues on missing metadata and missing properties in metadata (either mandatory or recommended). Open issues can be used to encourage contributors to solve them. Improved metadata also improves Google Dataset Search's ability to locate the data sets.
- Link and group data: Each dataset repository can contain a link-set to other datasets. Linksets can be automatically generated by a linking tool integrated into the portal or be contributed by dataset consumers. Contributions by data consumers will be managed through the version control infrastructure. Linksets can then be consumed by clients or loaded into a SPARQL endpoint to enable queries and data stories. This is a crucial ability as very few portals currently enable this linking as measured by the 5 Stars of Linked Open Data, and yet it is probably the most well-known metric for assessing data.

We advocate that CDSs need to operate in two different modes of operation: A first mode, that we call *dataset-centred,* is set around a single dataset, or a collection of related datasets owned by the same organisation. The organisation has control of the CDS and the inclusion of other datasets is in principle not allowed, limiting activity to the management of releases, distributions, derivations and quality improvements suggested by the community. This mode is meant to align with the needs of data owners. The second mode, that we call *re-use-centred*, is set around the realisation of a use-case that may require several datasets from different sources. For example, one could imagine CDSs being created for "Demographic data in Spain", or "Police spending across Europe", where besides grouping and cleaning datasets, a community develops code and/or visualizations related to the use case they want to realise. This mode is meant to foster the creation of communities. Both modes should have almost the same technical functionalities to reduce the complexity of deployment.

We implemented a prototype of a community data space on top of the open source Gitea platform.[4] Compared to other open source alternatives, we deemed Gitea the most advanced and better maintained of the lightweight platforms. Compared to GitLab, the most well-known open source DVC system, we considered Gitea easier to extend. We envision that community data spaces could come in two flavours:

[4] https://gitea.io/

a lightweight based on Gitea for self-hosting communities, and a full-fledged port to GitLab that could be used by portals and metaportals to provide hosting, taking advantage of GitLab's cloud support.

4. Use case examples: Publishers and users

Following Gitea's model, our modified version is packaged as a docker container that can be installed on any server with the docker Engine (>=18.0) and the docker-composer tool (>=1.25). To illustrate the functionalities of the prototype, we will describe the journey of Bob, the owner of the dataset of the street names of the Italian region of Alto Adige[5], and Alice, a member of the public interested in re-using it, and their interaction around a data-centred CDS. We assume both Bob and Alice have already created accounts in the system.

Bob creates a data space (Figure 4), just as if it was a repository on a DVC, with an unique name and a description. We also chose to keep the original Gitea functionalities of allowing private repositories and custom presentation templates. The data space is comprised of a list of tabs, each of them implementing a different functionality. Some tabs are unchanged from the underlying Gitea codebase, while others have been added or modified by us.

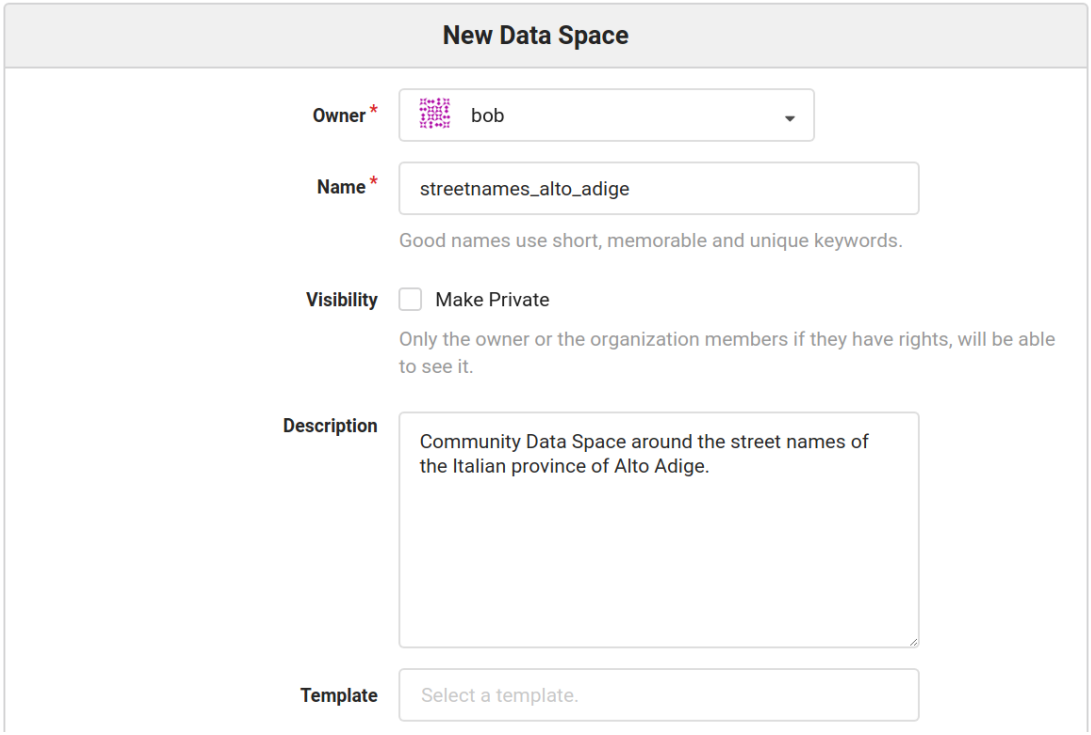

Figure 4: Creation of a data space in our prototype

We first describe the "Explore" tab, that shows the Git repository that comes with the data space (Figure 5). Bob can clone the repository in his local machine, and using git commands, add and commit datasets

[5] https://www.europeandataportal.eu/data/datasets/p_bz-streetnames-from-addresspoints

and metadata and publish it to the data space. He may also use the "New File" button to directly create a text file, or the "upload file" button to upload from his local computer.

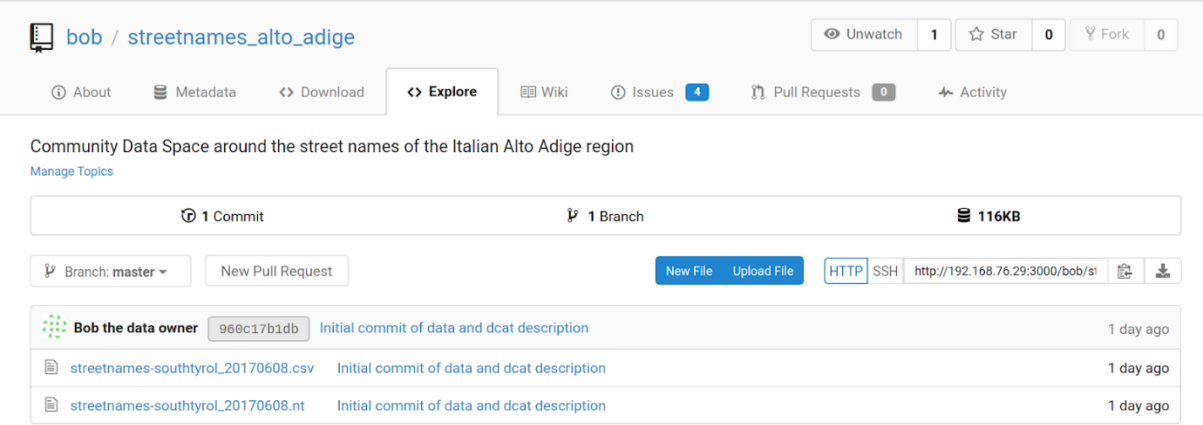

Figure 5: The "Explore" tab provides an overview of the Git repository of an example community data space from our protype

We modified Gitea's original "About" to show the DCAT description of the dataset in a tabular format, following how data portals present it (Figure 6). This modification is more appropriate for dataset centred CDSs. re-use centred CDSs may prefer to revert to the original behaviour of this tab, or, if CDSs get their own DCAT-style specification, use it instead.

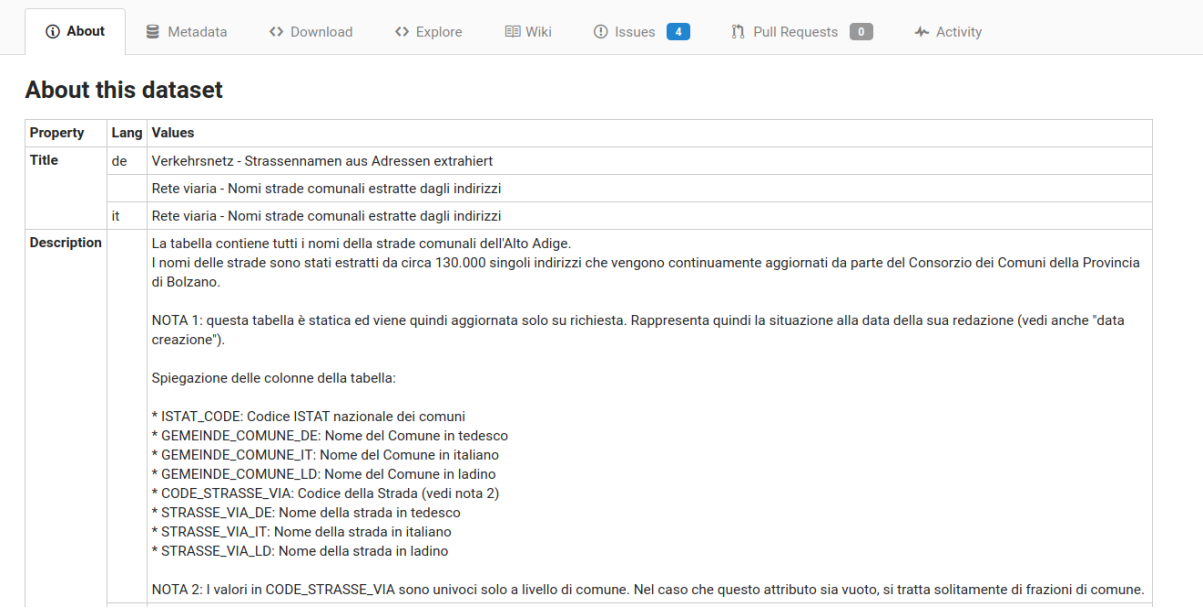

Figure 6: The "About" tab showing DCAT description of a dataset in our prototype

Bob sees the "About" tab and realises that the title and description of the dataset is only available in German and Italian. He thinks that it would be useful to translate the titles to other European languages to facilitate re-use, but he only knows German and Italian. To draw the attention of the community, he opens and issue through the "Issues" tab. The "Issues" tab is largely unchanged from Gitea's original. Issues can be assigned labels that can be configured on a per-CDS basis. Software developers commonly use labels such as "Bug", "Feature Request", "Deprecated", among others intimately related to their

particular software development methodology. Re-use centred CDSs may follow a similar approach for the code they develop inside CDSs. For datasets, both in re-use and dataset-centred CDSs we proposed the following set of default labels:

- *Metadata*: issues concerning the metadata of datasets
- *Quality issue*: issues concerning data quality, e.g., incompleteness or inaccuracy.
- *Linking*: issues and improvements concerning linksets and linking to other datasets
- *Derivations*: for proposals and discussions around derived datasets in the CDS
- *Format/Mappings*: for proposals of distributing the dataset in other formats, including the development of mappings.

Figure 7 shows the issue that Bob opened asking for help with the translation of descriptions.

Alice finds the CDS of the dataset by using the search box of the system. In our current prototype, we did not change Gitea's original search functionality, based on matching text with repository descriptions. We envision to replace this with metadata-based search like the one currently used in the EDP. We would rely on the schema.org markup functionality already implemented on current dataset pages to make CDSs discoverable by commercial search engines.

Alice clones the Git repository of the CDS and analyses the data. Right away, she notices that she has a couple of issues. She does not understand very well what a certain column represents. She also notices that there is a street missing in the dataset. She then opens one issue for each problem. Figure 8 shows the list of issues from the "Issues" tab after Alice logs her issues.

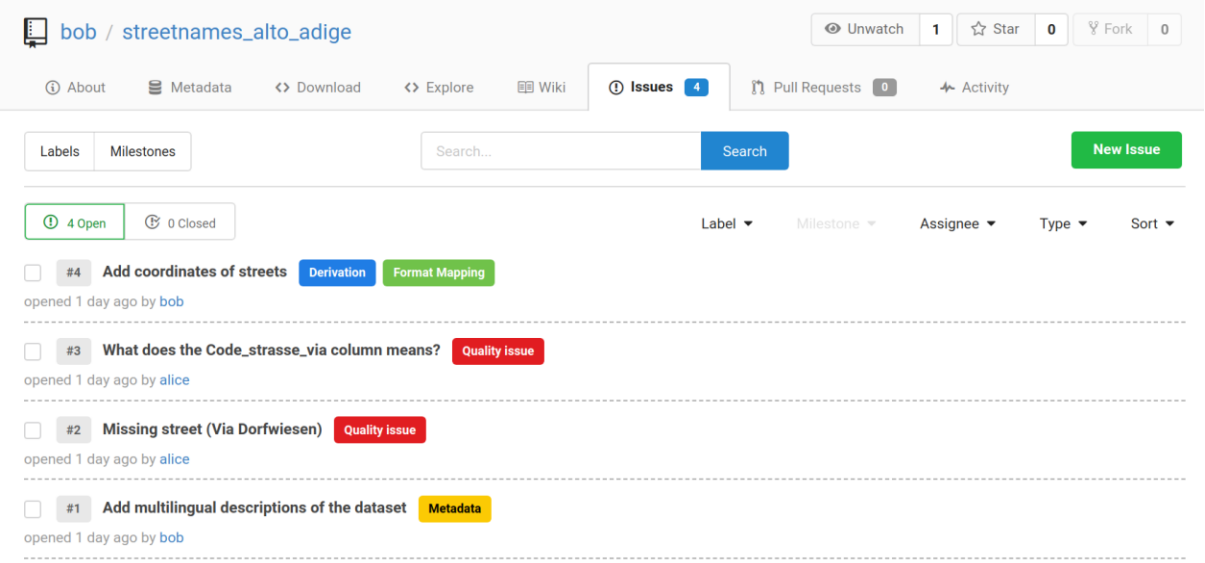

Figure 7: List of issues in the example CDS from our prototype

From the list, Alice notices Issue #1 about multilingual descriptions opened earlier by Bob. Alice happens to be able to contribute with a Spanish translation of the title and description and is eager to do it. Following from the DVC model, she has three possible courses of action:

1. She could provide the translation as a comment or attachment on the issue. If approved, Bob can update the file in the repository himself.

2. She could ask Bob to be included as a collaborator in the data space, giving her writing privileges on the repository. then, she can do the changes on her local copy and commit them to the repository using Git.
3. She could "fork" the data space, creating a new one, owned by her, that includes a copy of the repository. The "fork" retains the link to its parent repository, allowing Alice to create a "Pull request" with her changes. This "Pull request" is then reviewed by Bob, that has the final decision on accepting or rejecting it.

Options 1) and 3) are more appropriate for dataset centred CDSs, as they give more control to dataset owners about what changes generate a new "official" version of the dataset. Option 2) could also be used by re-use centred CDSs, where the benefits of having more people readily able to contribute outweigh the potential issue of temporarily having incorrect versions in the repository[6].

Let's see how this would look if Alice decides to follow option 3). First, she creates a fork by clicking on the link on the upper right corner of the data space page (Figure 9). From the newly created data space, she clones the Git repository in her local machine, adds the new title to the DCAT description and commits the changes (Figure 10). She is now ready to create a "Pull Request" to the original data space created by Bob using the tab on the upper menu (Figure 11). Bob receives a notification that a pull request has been logged, he can review the changes, initiate a discussion, and if satisfied, approve the changes and "merge" the request into the repository (Figure 12).

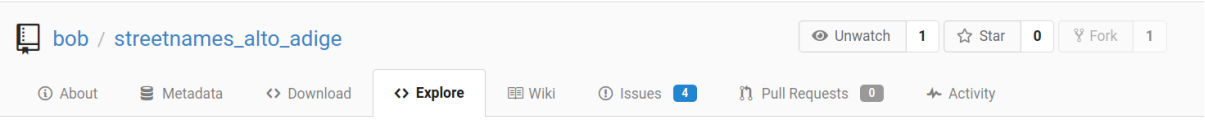

Figure 8: The link to create a fork is on the upper right corner of the data space page

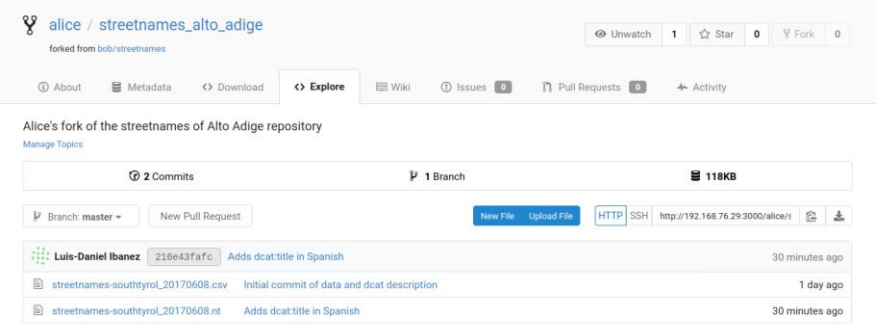

Figure 9: View of Alice's forked data space. She just modified the metadata file to add the Spanish translation of the title (bottom of the figure)

[6] However, remember that thanks to the use of DVC all changes can be reverted. If Alice commits a mistake to the repository, Bob can easily revert to the previous version. If Alice turns out to behave in bad faith, her collaborator status can be revoked.

At this stage, we have been able to leverage the capabilities of Gitea to enable collaboration around datasets. However, we do assume that Alice knows how to use Git. While this is a reasonable assumption for software developers and data scientists, we would also like to integrate non tech-savvy users into CDSs, in line with the principle of "foster the participation of citizens" outlined in the EU's open data policy[7]. Non tech-savvy users can still collaborate by creating issues and/or participating on them, but to further advance towards their inclusion, we added a "Metadata" tab (Figure 13), that allows a logged user to edit the metadata of a dataset directly on the browser, without the need for Git. Behind the scenes, a pull request with the changes made is created on behalf of the user[8], that can be reviewed in the same way as one created like Alice did in our example. This approach may also be used to modify datasets for which an in-browser editor exists.

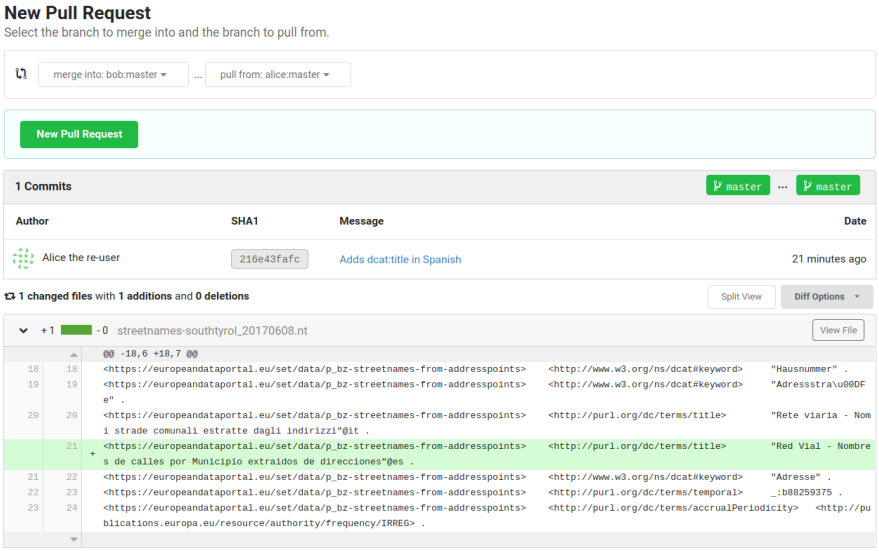

Figure 10: Alice creating a pull request to the parent repository "bob:master" from her repository "alice:master". A recap of the changes is also available.

[7] https://ec.europa.eu/digital-single-market/en/open-data

[8] Note for the Git-savvy: There are two ways of implementing this: (1) temporarily add the user as collaborator, create a branch with their changes, create a pull request based on that branch, remove the user as collaborator; (2) have a system user with writing permissions to all data spaces and use it to create the pull requests, including a note indicating the user that originated it. Both have pros and cons that are beyond the scope of this report.

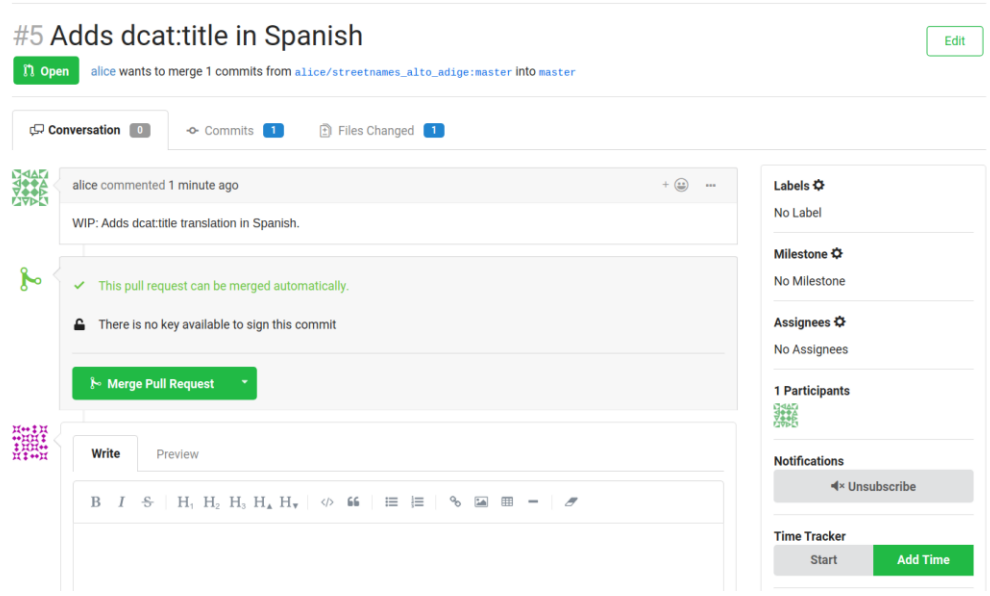

Figure 11: Bob's view of Alice's pull request. He can initiate a discussion, approve the request by clicking on "Merge Pull Request", or review the changes by clicking on "Files Changed"

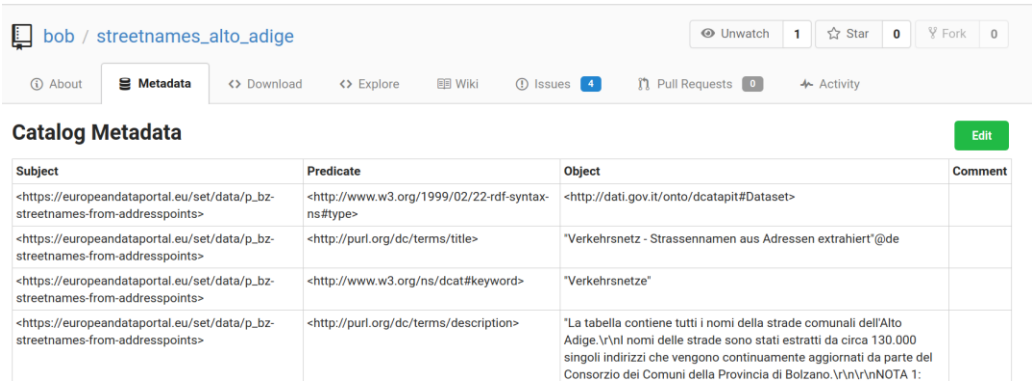

Figure 12: "Metadata" tab enables edition of DCAT metadata by non-technical members of the community.

We also modified Gitea's "Download" tab (Figure 13) to provide direct links to the different distributions (in the DCAT sense) on which datasets are available, reproducing similar functionality of open data portals. This tab makes more sense for dataset centred CDSs and can be deactivated via configuration. The final tab of our prototype is the "Wiki" tab that provides a CDS with a wiki-style forum and is currently unchanged from what Gitea provides.

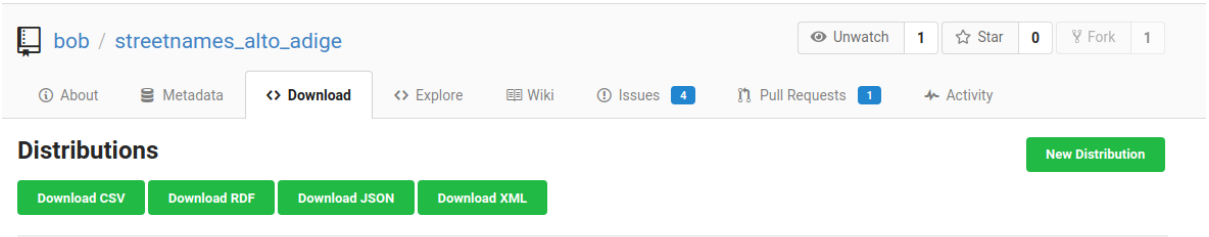

Figure 13: Owner's (Bob) view of the "Download" tab

5. Pilot

With the engineering feasibility of the approach confirmed, the next step is to conduct a pilot of the approach, designed to evaluate if collaborative data spaces actually represent an improvement for their intended end users: data owners and data re-users, so to justify the infrastructure investment needed to kickstart the approach. For that, user studies should be undertaken to assess:

- As a re-user:
    - Does a dataspace community around a dataset facilitate how I re-use a dataset?
    - Provided that I'm willing to contribute in a dataspace, does it allow me to do so effectively?
- As a data owner:
    - Does a dataspace facilitate the measuring of re-use with respect to current situation? Does it help with making datasets more re-usable?

User studies would also provide hints about if there are problems with the concept of dataspace, if there are technical and usability shortcomings that could be solved, or if there are further use cases that need to be considered.

From re-users' perspective, a pilot study could be run by recruiting data enthusiasts already engaged in re-using open data and study their interactions with the prototype both for the case of re-using and improving a single dataset, and for the case of kickstarting a community around several datasets. From the owners' perspective, the study should engage with them and present them with a simulation of an active dataspace around datasets they own, to answer: how would it integrate with their current workflows for managing datasets?

Such a pilot would also give insight on how one could build a general portal community. For example, if the pilot runs at a higher level portal where the EDP sits, one could measure how different data spaces overlap in terms of what datasets are used by most of them, what members join multiple data spaces, and which of the co-located tools are used more often. A pilot including multiple data spaces would also help to assess the factors that characterize a successful community.

6. Recommendations

Some lessons learned and best practices can be pointed out:

- For dataset-centred CDSs, define conventions for the layout of a dataset repository, including formats, metadata and documentation. The convention would be encouraged by being automatically instantiated every time a repository is created.
- Hosting CDSs in a portal requires an extra investment in storage, as datasets need to be copied to enable full functionality. In terms of computing power, needs are a function of the number of expected users and of the requirements of other tools that a portal would like to co-locate. When describing the metadata edition functionality, we mentioned the possibility of embedding in-browser editors to reduce the need of Git knowledge to be able to collaborate. This can be

extended to the provision of more complex tools like Open Refine (for curation) or Jupyter Notebooks (for visualization)

- Big datasets are more difficult and costlier to include in this paradigm. Git-LFS enables the offloading of big files to a separate server at the expense of version control granularity. If one expects to handle several versions of the same big datasets, storage would need to be allocated accordingly. In the context of the EDP the great majority of very large open datasets are geospatial or satellite images, for which there is not an easy way to link beyond metadata. This suggests that Geoportals may need a different implementation of CDSs, with appropriate infrastructure and their own set of specialised tools.
- Data that is accessed by an API (notably, streaming data) won't take advantage of versioning capabilities, however there is still room to co-locate ways to access the data (call the API), and its usage.
- We offered a partial solution to the assumption that Alice the data re-user needs to be knowledgeable in Git, but Bob the data owner also needs to have at least basic notions to manage a data space effectively. If our proposed dataset-centred operation mode were to succeed, we need to guarantee that this is not a barrier for data owner organisations.

7. Conclusion and future work

After feedback from end users has been collected, there are a number of interesting avenues for future work. From a technical perspective, it is important to better integrate data and services that are accessed through APIs and quantify the added infrastructure cost.

More in the medium-long term, as CDSs evolve from Minimum Viable Products to full-fledged systems, an interesting direction is to pinpoint if the Git-style workflow is sufficient for the open government data context, or if specific commands and primitives are required. Some Machine Learning practitioners consider that pure Git is not enough for version control of learning models and the datasets used for training and verification. They have developed their own version controller called DVC[9], extending Git to include lightweight pipelines as a first-class citizen. DVC introduces a new primitive 'dvc repro', to reproduce experiments end-to-end, an important use case for this community. Do CDSs require a similar adaptation? Or perhaps using DVC instead of pure Git would be enough?

From a social perspective, do we observe the same patterns and effects observed in open source software with datasets? Can we characterize the differences and what do they tell us about how to improve the interaction of re-users with datasets? In this first implementation, a pre-condition for use is to be familiar with distributed ver, how to include also non-technical savvy citizens in order to foster their participation in political and social life as aimed by the Open Data policy of the commission? [10]

[9] https://dvc.org DVC stands for "Data Version Control" and should not to be confused with "Distributed Version Control"

[10] https://ec.europa.eu/digital-single-market/en/open-data