Project Name	**FREYA**
Project Title	**Connected Open Identifiers for Discovery, Access and Use of Research Resources**
EC Grant Agreement No	**777523**

D2.2 PID Metadata Provenance

Deliverable type	Other
Dissemination level	Public
Due date	31 May 2019
Authors	Martin Fenner (DataCite, https://orcid.org/0000-0003-1419-2405) Joe Wass (Crossref, https://orcid.org/0000-0002-0840-454X) Tom Demeranville (ORCID, https://orcid.org/0000-0003-0902-4386) Sarala Wimalaratne (EMBL-EBI, https://orcid.org/0000-0002-5355-2576) Richard Hallett (DataCite, https://orcid.org/0000-0002-8599-0773)
Abstract	The implementations of provenance tracking for metadata by the persistent identifier (PID) providers in the FREYA project are described.
Status	Submitted to EC 31 May 2019

The FREYA project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 777523.

FREYA project summary

The FREYA project iteratively extends a robust environment for Persistent Identifiers (PIDs) into a core component of European and global research e-infrastructures. The resulting FREYA services will cover a wide range of resources in the research and innovation landscape and enhance the links between them so that they can be exploited in many disciplines and research processes. This will provide an essential building block of the European Open Science Cloud (EOSC). Moreover, the FREYA project will establish an open, sustainable, and trusted framework for collaborative self-governance of PIDs and services built on them.

The vision of FREYA is built on three key ideas: the **PID Graph**, **PID Forum** and **PID Commons**. The PID Graph connects and integrates PID systems to create an information map of relationships across PIDs that provides a basis for new services. The PID Forum is a stakeholder community, whose members collectively oversee the development and deployment of new PID types; it will be strongly linked to the Research Data Alliance (RDA). The sustainability of the PID infrastructure resulting from FREYA beyond the lifetime of the project itself is the concern of the PID Commons, defining the roles, responsibilities and structures for good self-governance based on consensual decision-making.

The FREYA project builds on the success of the preceding THOR project and involves twelve partner organisations from across the globe, representing PID infrastructure providers and developers, users of PIDs in a wide range of research fields, and publishers.

For more information, visit www.project-freya.eu or email info@project-freya.eu.

Disclaimer

This document represents the views of the authors, and the European Commission is not responsible for any use that may be made of the information it contains.

Copyright Notice

Executive summary

The main focus of this document is describing the implementations of provenance tracking by the persistent identifier (PID) providers Crossref, ORCID, Identifiers.org (EMBL-EBI), and DataCite in the FREYA project. We describe the implementations by Crossref and ORCID done previously, the DataCite implementation done as main output for this deliverable, and the conceptual work by identifiers.org.

Crossref, ORCID and DataCite provide persistent identifiers with metadata, so provenance considerations are about the provenance of metadata rather than the provenance of the research outputs that the metadata describe. Identifiers.org provides globally unique persistent identifiers but no metadata about the content described by these persistent identifiers, so the need for provenance tracking is more limited.

In April 2019 DataCite launched its provenance service as a new Activities API, which for the first time tracks activities around DataCite DOI metadata and makes this information publicly available. Since the service started, close to 10 million activities have been tracked and the service is operating as part of the DataCite production infrastructure.

The implementations by Crossref, ORCID and DataCite are all based on the PROV-DM data model, but differ in their implementation, and the terminology used. The main reasons for that are the major differences in the provenance information that needs to be tracked for DataCite DOI registrations, assertions in ORCID records, and external information linking to DOIs in Crossref/DataCite Event Data.

Provenance is a key topic in FREYA, both in this Work Package (WP2) and in WP4 (Integrating the PID Graph). While the work in WP2 is concerned with provenance of persistent identifier metadata, the work in WP4 focuses on sharing provenance information about resources or the metadata of the resources and their relations with other resources.

Contents

1 Introduction 5
2 Existing implementations 7
2.1 Crossref 7
2.1.1 Event Schema 7
2.1.2 Crossref Evidence Logs 8
2.1.3 Blended approach 9
2.2 ORCID 9
2.2.1 Mapping PROV to ORCID 9
2.3 Identifiers.org 11
3 Implementing metadata provenance for DataCite DOIs 14
3.1 Use cases 14
3.2 Service architecture 14
3.3 API overview 14
3.4 Example 15
3.5 Tracking changes 16
3.6 Result representation 16
3.7 Initial feedback and next steps 16
4 Conclusions and future work 18

1 Introduction

Provenance, understood here as the systematic management of the records of origin of research artefacts, is an important aspect of Open Science, as it provides contextual information of how and from what sources research originates. Provenance contributes to FAIR principles (Findable, Accessible, Interoperable, Reusable) as it can facilitate research reusability and reproducibility.

“Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness.” PROV-DM

PROV-DM is the conceptual data model that forms a basis for the W3C provenance (PROV) family of specifications.[1]

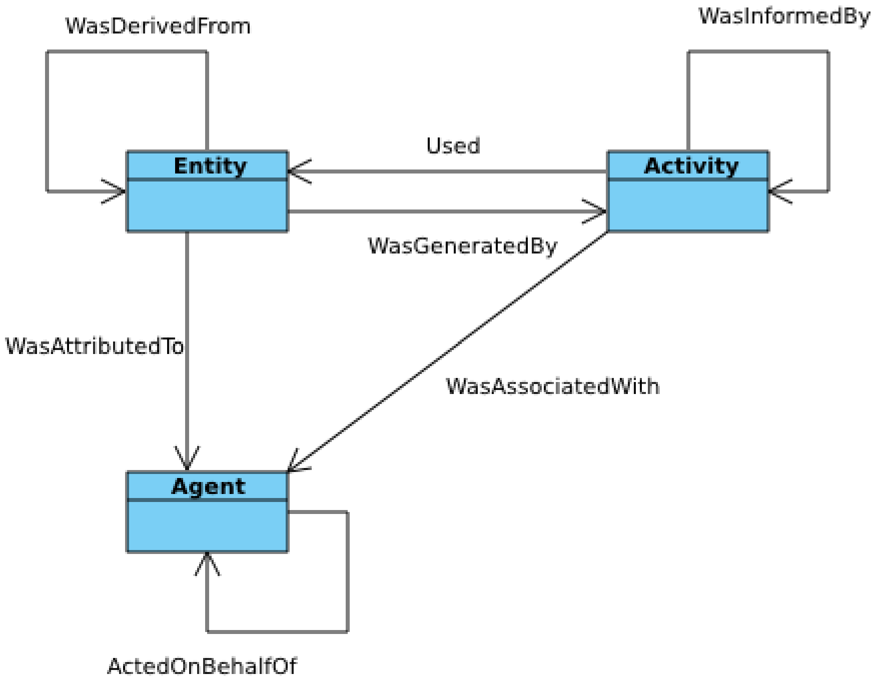


Figure 1 Basic PROV concepts

The PROV core concepts (Figure 1) include the types entity, activity and agent, and the following relations:

1. entity-activity

   WasGeneratedBy

   Used

   WasInformedBy

2. entity-entity

   WasDerivedFrom

3. entity-agent

   WasAttributedTo

[1] https://www.w3.org/TR/prov-dm/

WasAssociatedWith

ActedOnBehalfOf

With regard to persistent identifiers (PIDs), there are different flavours of provenance that can be discerned: provenance of persistent identifiers themselves and their associated metadata as specific research artefacts, and provenance of other research artefacts with PIDs contributing to making clear statements about the artefacts' origin and connections to other PIDs. Provenance is a core concept for creating connections in the PID Graph to contextualise content persistently. The metadata associated with PIDs helps enrich the PID Graph with the necessary information to support a resource's identity (e.g. contributor/s, production date, etc.), which supports trust.

The persistent identifier (PID) service providers in the FREYA project (DataCite, Identifiers.org, Crossref, ORCID) provide metadata, so provenance considerations are about the provenance of metadata rather than the provenance of the research outputs that the metadata describe. Two of the three basic types in the PROV data model (entities and actors) are well described by PID metadata, but activities are typically not tracked specifically.

This document describes the work in the FREYA project on PID metadata provenance. The primary output is a new provenance service by DataCite launched to production (TRL 8) in April 2019. This document describes this new service, the extensive work done by ORCID and Crossref before work on this deliverable started, and the lessons learned from implementing PID metadata provenance in multiple PID services.

Related work by the disciplinary partners in the FREYA project is described in deliverable D4.2 and focuses on the provenance of the artefacts described by persistent identifiers and metadata.

2 Existing implementations

2.1 Crossref

Crossref has implemented detailed tracking provenance in Event Data[2]. Event Data is a collaborative project between Crossref and DataCite, and is production-ready since 2018[3]. When a relationship is observed between a registered content item (that is, content that has been assigned a DOI by Crossref or DataCite) and a specific web activity, the data is expressed in the service as an **event**. The event schema, which is common to both organisations, records basic provenance. Further, it allows each event to link to more fine-grained provenance information. Crossref provides this level of detail for events it has generated in **Evidence Logs**[4] and **Evidence Records**[5].

PROV-DM	Crossref Evidence Records
Entity	Event
Activity	Action
Agent	Agent

Table 1 Mapping Evidence Records to PROV-DM types

2.1.1 Event Schema

The Event schema, which was derived from the Lagotto project[6] by the open access publisher PLOS, funded by the Arthur P. Sloan Foundation, provides the following provenance fields:

1. **source** - Compulsory. This records the original location of the data.
2. **source token** - Compulsory. This records the specific Agent that collected the data and produced the event.
3. **evidence record** - Optional. URL that points to a machine-readable document describing how the event was produced.

An example evidence record is described here.

Event Data contains a wide variety of data sources and paths. As such, the **source** field works at a number of degrees of abstraction. Some example **source** values:

1. **reddit** - The data was derived from content on the Reddit platform.
2. **twitter** - The data was derived from content on the Twitter platform.
3. **crossref** - The data was derived from Crossref metadata, as supplied by Crossref members. Crossref may perform some cleaning up (e.g. reference matching) as part of this process.
4. **datacite** - The data was derived from the DataCite metadata, as supplied by DataCite members.

There is no single point at which the chain of provenance can be said to end, but we can draw a line at which provenance ceases to be useful. For example, the inner workings of Twitter are not relevant, nor

[2] https://www.crossref.org/services/event-data/
[3] https://www.crossref.org/blog/event-data-is-production-ready/
[4] https://www.eventdata.crossref.org/guide/service/evidence-logs/
[5] https://www.eventdata.crossref.org/guide/data/evidence-records/
[6] https://github.com/lagotto/lagotto

visible. The source in Event Data therefore describes the most significant boundary over which the data has crossed before entering Event Data.

It is possible for data to come from a given **source** via two routes. For example, it's possible that two agents might collect data from Twitter and process it in different ways. The s**ource token** identifies a particular agent that consumed data from the **source** and produced the event. Each piece of agent software is assigned a source token which can then be used to trace the event back to that service.

2.1.2 Crossref Evidence Logs

Data is ingested from a source, producing events. Along its journey there are many factors that affect the production of the event. These include different entities, processes and agents. For example:

1. The Twitter API entity supplies data. This API has certain characteristics that a consumer might like to know about.
2. The Twitter API sends data based on a set of filters. This entity determines which data is sent.
3. The Crossref Twitter agent extracts the data. The behaviour of this agent determines how events are produced. The agent is versioned software, and the versions change over time.
4. The Crossref Percolator is an agent that extracts links to produce Events. This is versioned software whose behaviour can change over time.
5. Various artifacts, such as the list of domain names and DOI prefixes are consumed as an event is created. These entities are versioned.

In addition to this, environmental factors such as the availability of various APIs or interconnections, may affect the production of data.

All of the above are documented in two Entities:

1. Evidence Record, which is a JSON document with a given identifier. This describes, in a declarative fashion, which data came in and out, and how it was processed.
2. Evidence Log, which is a stream of structured logs that describes the activities and each decision point. The logs contain identifiers of Evidence Records for traceability.

Crossref Events fetch data from external sources in batches. Each batch is represented in an Evidence Record, which may produce zero or more events. Each event contains a link to the relevant Evidence Record that describes the activity that produced it.

The Evidence Record contains:

1. ID and version of the agent software that fetched the data.
2. ID and version of the Percolator software that processed the data.
3. Versions of any artifacts that were involved in the collection or processing of the data.
4. Input data and types (e.g. blog URLs)
5. Candidate matches (e.g. URLs found in content that may link to an article)
6. Matches (e.g. URLs matched to DOIs)
7. Events produced
8. Method used for matching the URL to the DOI, and to verify the association.

The Evidence Log is a series of lines which contain:

1. Evidence Log ID
2. Timestamp of activity.
3. Activity ID. These are listed in the documentation. e.g. "I tried to match a URL to a DOI"
4. Result of activity, e.g. "Failed" or "Succeeded, got DOI X"

2.1.3 Blended approach

The provenance of a given piece of information in Event Data is not only detailed, but the type and depth of detail is diverse across the types of data. There is no one-size-fits-all solution to recording provenance, and multiple audiences who may be trying to answer different types of questions. The blended approach includes source name, ID of agent, version of software, decisions taken, data ingested, problems encountered.

2.2 ORCID

The ORCID Registry enables connections between individuals (via their ORCID iD) and their activities and affiliations (via other identifiers and APIs). These connections are asserted by individual record-holders themselves, or by organizations they interact or otherwise are affiliated with. Providing transparent information about how assertions are added to the registry is key to maintaining and building upon trust extended to ORCID by the research community.

For ORCID, an assertion is a statement that connects a person, with an item, and includes information about the source (the agent that used the API to make the update). Assertions are entities in PROV-DM terms. In addition to a person, item, and source, an assertion may also include the 'assertion origin' (the agent that made the connection between item and person) where this agent is different from the source. We also consider the concept of item origin (the agent responsible for the item origin - e.g. a journal).

ORCID makes it clear who added an assertion to a record in both the UI and API metadata. Simple examples, currently supported by ORCID, include:

1. Assertion by Person: A person enters their employment details directly into the ORCID user interface. The person is listed as the Source.
2. Assertion by Member: A person submits an article to a journal, which collects an authenticated iD and update permission. When the article is published, the journal uses the permission to update the person's ORCID record, which lists the journal as the source.

More complex examples that distinguish between who added an assertion and who made an assertion (currently being piloted by ORCID) include:

1. Assertion by Member on behalf of a Person: A person adds information to a profile system, or chooses information to add to their record from an indexer. The system is an ORCID member and updates ORCID with itself as the **source**, and the user as the **assertion origin**.
2. Assertion by Member on behalf of another Member: An ORCID member provides services to other members, and adds their assertions with itself as the source. The member it is working on behalf of is recorded as the **assertion origin**.

2.2.1 Mapping PROV to ORCID

PROV is useful when two PROV using systems wish to exchange provenance information with each other. It can be used, as shown in this document, to describe provenance-of-provenance problems which are the basis of the ORCID use case, i.e. who is stating that a particular person contributed to an output.

PROV does not cleanly map to the ORCID use case. It is missing our domain language we require to reason about ORCID records and has been supplemented here with these definitions. Specifically, the "wasAttributedTo" relationship is not specific enough for our needs and the "wasGeneratedBy" relationship fails to clarify when actions have more than one agent associated with them (we distinguish between who added/updated metadata and who decided what to add/update). This means that it is not possible to tell which is the Item **origin**, **assertion origin** or **source** by examining the included PROV diagrams, without it being specifically stated in footnotes or the diagram itself. It is possible that the diagrams could be expanded to include activities that represent ‘item upkeep’, ‘asserting’ and ‘authorising’ in order to more accurately (but still incompletely) capture this, but the results would be far more complex than they already are.

Figure 2 and Figure 3 show how we map ORCID’s domain concepts to the PROV model:

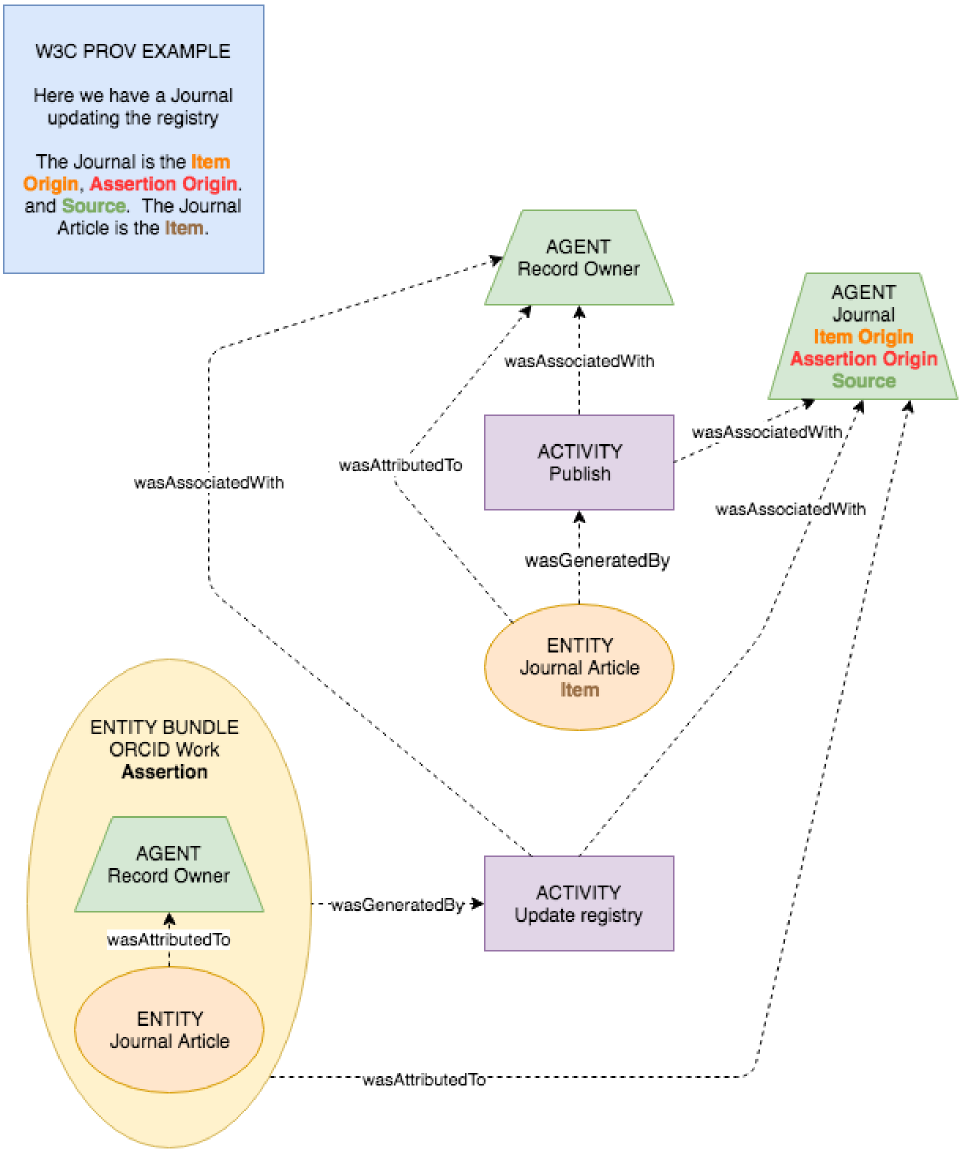


Figure 2 Mapping of ORCID concepts to PROV model

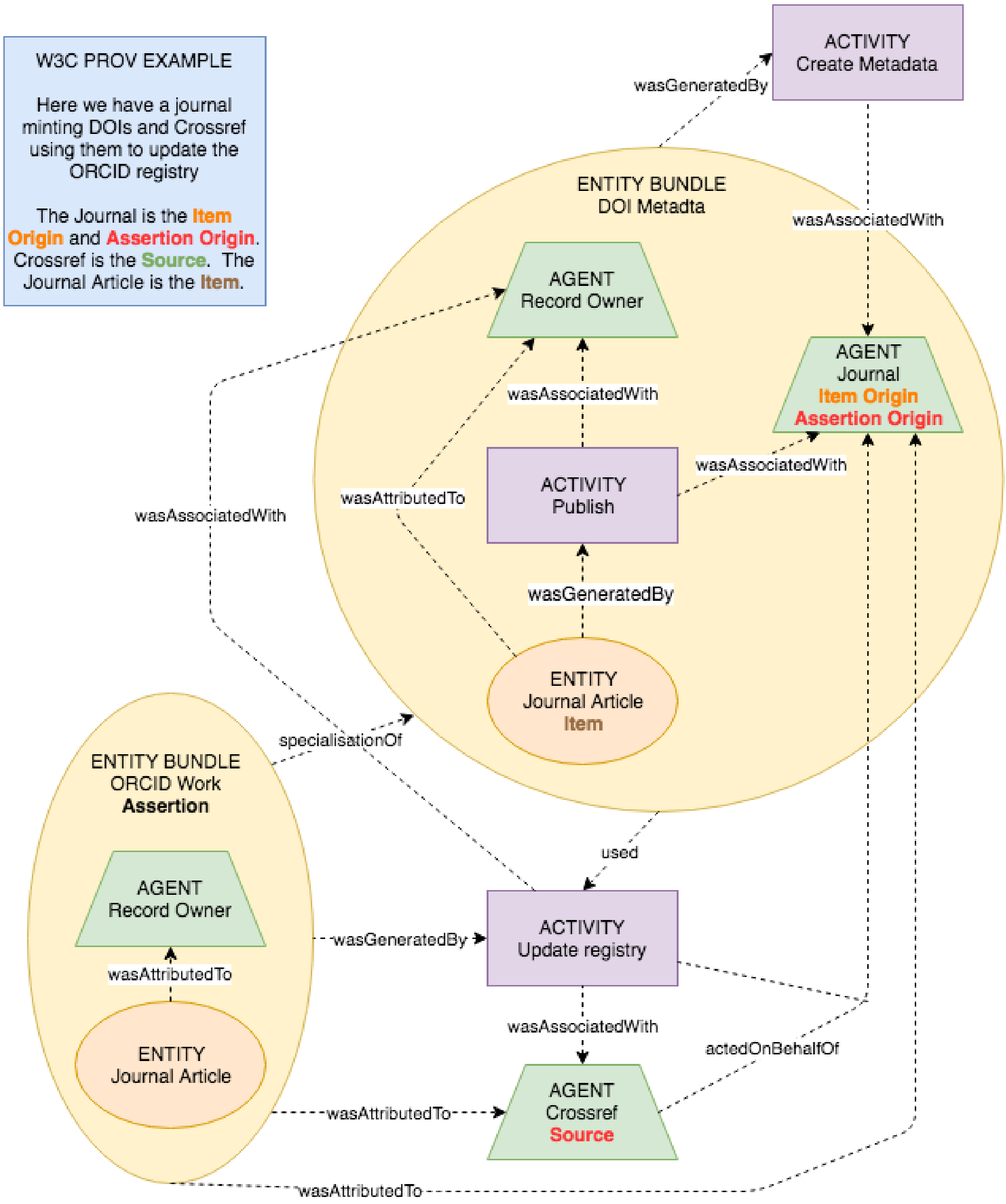
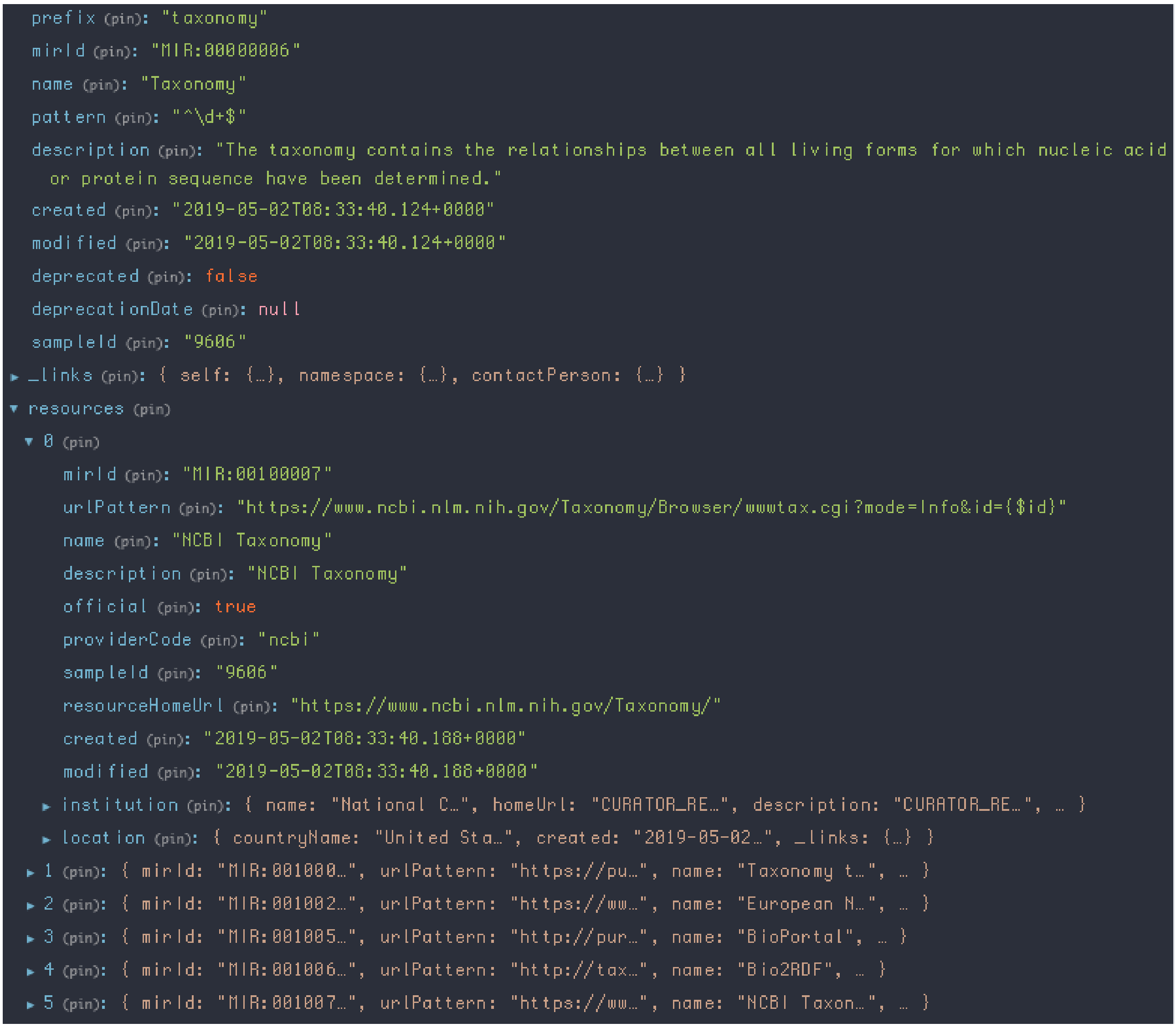


Figure 3 Mapping of ORCID concepts to PROV model

2.3 Identifiers.org

Identifiers.org assigns unique prefixes to data resources to construct globally unique identifiers. For example, Protein Data Bank is assigned ‘pdb’ as a prefix. This allows identifiers.org to uniquely identify pdb datasets in the form of compact identifiers, pdb:{local identifier or accession number} eg: pdb:4hhb.

During the prefix assignment process, we capture metadata about the resource (e.g. title, description, home URL, access pattern, primary resource etc) and create event (date created and creator). Figure 4 shows the information stored for the prefix **taxonomy**. This information can be accessed via https://registry.api.hq.identifiers.org/restApi/namespaces.



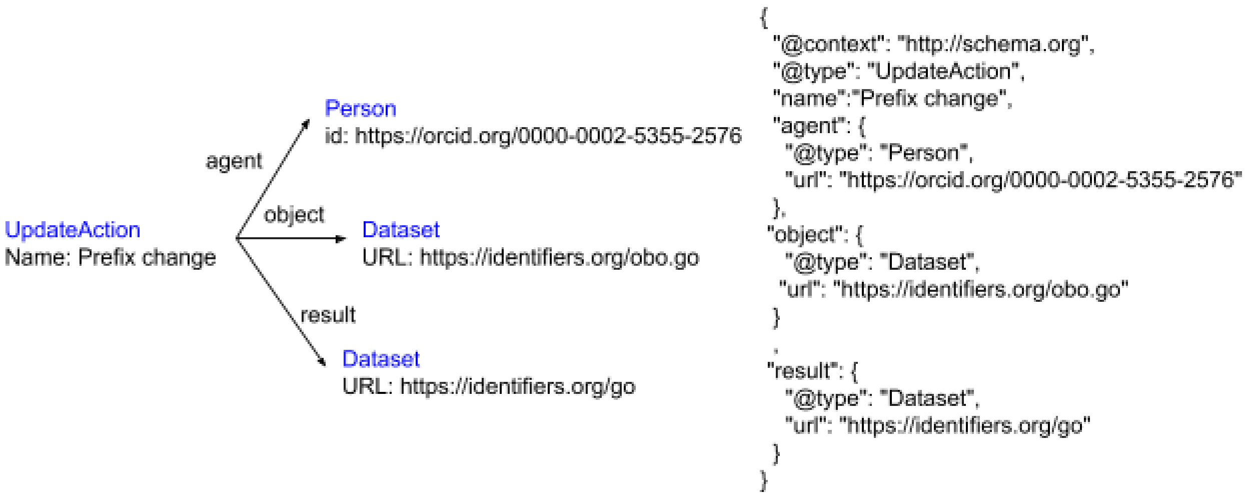
```
  prefix (pin): "taxonomy"
  mirId (pin): "MIR:00000006"
  name (pin): "Taxonomy"
  pattern (pin): "^\d+$"
  description (pin): "The taxonomy contains the relationships between all living forms for which nucleic acid
    or protein sequence have been determined."
  created (pin): "2019-05-02T08:33:40.124+0000"
  modified (pin): "2019-05-02T08:33:40.124+0000"
  deprecated (pin): false
  deprecationDate (pin): null
  sampleId (pin): "9606"
▸ _links (pin): { self: {…}, namespace: {…}, contactPerson: {…} }
▾ resources (pin)
  ▾ 0 (pin)
      mirId (pin): "MIR:00100007"
      urlPattern (pin): "https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?mode=Info&id={$id}"
      name (pin): "NCBI Taxonomy"
      description (pin): "NCBI Taxonomy"
      official (pin): true
      providerCode (pin): "ncbi"
      sampleId (pin): "9606"
      resourceHomeUrl (pin): "https://www.ncbi.nlm.nih.gov/Taxonomy/"
      created (pin): "2019-05-02T08:33:40.188+0000"
      modified (pin): "2019-05-02T08:33:40.188+0000"
    ▸ institution (pin): { name: "National C…", homeUrl: "CURATOR_RE…", description: "CURATOR_RE…", … }
    ▸ location (pin): { countryName: "United Sta…", created: "2019-05-02…", _links: {…} }
  ▸ 1 (pin): { mirId: "MIR:001000…", urlPattern: "https://pu…", name: "Taxonomy t…", … }
  ▸ 2 (pin): { mirId: "MIR:001002…", urlPattern: "https://ww…", name: "European N…", … }
  ▸ 3 (pin): { mirId: "MIR:001005…", urlPattern: "http://pur…", name: "BioPortal", … }
  ▸ 4 (pin): { mirId: "MIR:001006…", urlPattern: "http://tax…", name: "Bio2RDF", … }
  ▸ 5 (pin): { mirId: "MIR:001007…", urlPattern: "https://ww…", name: "NCBI Taxon…", … }
```

Figure 4 Information stored for prefix taxonomy

In rare occasions, the resource provider may request to change the assigned prefix. The Identifiers.org infrastructure supports redirection of the content seamlessly for compact identifiers using old and new prefixes. Currently, this information is not exposed to the users following a formal provenance model. As part of the FREYA PID metadata provenance, we explored ideas around capturing prefix update using the schema.org **UpdateAction** type (Figure 5).

UpdateAction	The act of managing by changing/editing the state of the object.
Properties	
agent	Driver of the action.
object	The object upon which the action is carried out.

result	The result produced in the action.


Person
id: https://orcid.org/0000-0002-5355-2576
agent
object
UpdateAction
Name: Prefix change
Dataset
URL: https://identifiers.org/obo.go
result
Dataset
URL: https://identifiers.org/go

```
{
  "@context": "http://schema.org",
  "@type": "UpdateAction",
  "name":"Prefix change",
  "agent": {
    "@type": "Person",
    "url": "https://orcid.org/0000-0002-5355-2576"
  },
 "object": {
    "@type": "Dataset",
   "url": "https://identifiers.org/obo.go"
  }
  ,
 "result": {
    "@type": "Dataset",
    "url": "https://identifiers.org/go"
  }
}
```

Figure 5 Illustrative use of “UpdateAction”

3 Implementing metadata provenance for DataCite DOIs

As output of the work in the D2.2 deliverable on PID metadata provenance, DataCite launched a new production service for tracking provenance information about DOI metadata held by DataCite in March, 2019[7]. This service tracks changes made within the system for creation, updating or in some cases deletion of the metadata associated with DOIs. These changes are tracked as activities and exposed via new API endpoints within the existing DataCite infrastructure. The service started tracking DOI metadata changes March 10, 2019, and as of May 15, 2019 it has collected more than 8.3 million activities.

3.1 Use cases

The main use case for the activities API is to track all changes to DataCite DOI metadata to provide full transparency over any changes over time. This includes cases where something might have gone wrong, e.g. information accidentally overwritten, or changes to metadata that can not be easily explained. This means that activities information is important to have available, but will actually be used only rarely. The activities API increases the trust by members and users in information provided by DataCite services.

3.2 Service architecture

Prior to the work on this deliverable, the DataCite data model supported the PROV core concepts types **entity** (e.g. a dataset) and **agent** (e.g. a member), but could not describe **activities**. The first goal in designing the DataCite activities API was therefore to describe, store and expose **activities**.

Specifically, the goal was to identify mature existing Open Source solutions that provide this functionality. Of the several available solutions that integrated with the Ruby on Rails software stack that currently powers our APIs, we picked the **audited** gem[8], a mature and widely used implementation for tracking changes in a database backend for a Ruby on Rails application. We added an Elasticsearch index and JSON REST API for these activities.

3.3 API overview

The DataCite REST API is accessible via https://api.datacite.org/ - This is both a public and private API, the public side is the most relevant here for exposing all the provenance information. More information about the DataCite REST API can be found at on DataCite support pages: https://support.datacite.org/docs/api- This also includes relevant documentation related to this new provenance-based activities API. The API exposes endpoints for gathering and querying this data:

1. All activities https://api.datacite.org/activities - All the information about the activity events that have been recorded.

2. All activities filtered by a query https://api.datacite.org/activities?query=changes.url:* - The query parameter can be used to refine the activities to look for.

3. All activities related to a specific DOI https://api.datacite.org/dois/10.5438/wy92-xj57/activities - Only activities related to a specified DOI.

4. Specific activity - https://api.datacite.org/activities/5e4e59d8-cc3a-4017-9066-2f04b64aedb9 - If a activity ID is known, it's details can be directly retrieved.

[7] Fenner, M. (2019, April 10). Exposing DOI metadata provenance. https://doi.org/10.5438/WY92-XJ57
[8] https://github.com/collectiveidea/audited

3.4 Example

https://api.datacite.org/activities/bc5314df-da33-46c3-8828-4a50d1106252

```
{
  "data": {
    "id": "bc5314df-da33-46c3-8828-4a50d1106252",
    "type": "activities",
    "attributes": {
             "prov:wasGeneratedBy": "https://api.datacite.org/activities/bc5314df-da33-
       46c3-8828-4a50d1106252",
             "prov:generatedAtTime": "2019-04-15T12:23:24.522Z",
             "prov:wasDerivedFrom": "https://doi.org/10.5281/zenodo.2640674",
             "prov:wasAttributedTo": "https://api.datacite.org/clients/cern.zenodo",
             "action": "update",
             "version": 2,
             "changes": {
                     "url": [
                              null,
                              "https://zenodo.org/record/2640674"
                     ],
                     "aasm_state": [
                              "draft",
                              "findable"
                     ]
          }
    },
    "relationships": {
             "doi": {
                     "data": {
                              "id": "10.5281/zenodo.2640674",
                              "type": "dois"
                     }
             }
    }
  },
  "included": [...]
}
```

The API response also includes the metadata of the associated DOI(s).

In this example, the URL for the DOI was changed from “null” to “https://zenodo.org/record/2640674” and the DOI state from “draft” to “findable”. These are the expected changes in step 2 of DOI registration (where the DOI is registered in the handle system, step 1 is metadata registration). The provenance information in this API response is shown in Figure 6:

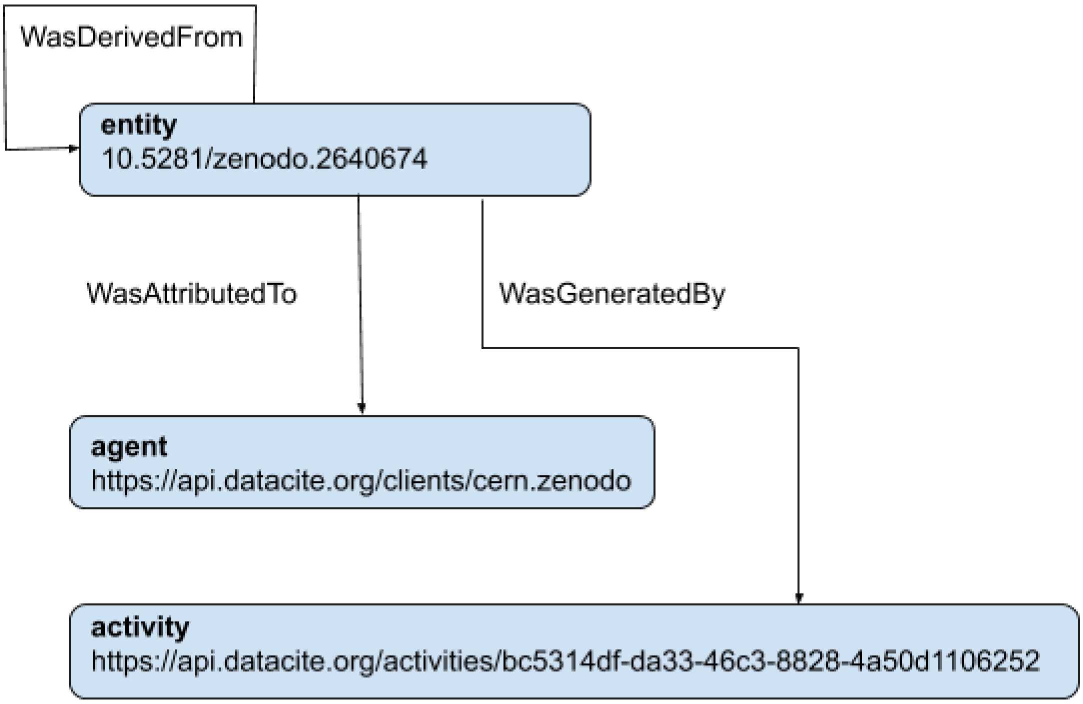

WasDerivedFrom
entity
10.5281/zenodo.2640674
WasAttributedTo
WasGeneratedBy
agent
https://api.datacite.org/clients/cern.zenodo
activity
https://api.datacite.org/activities/bc5314df-da33-46c3-8828-4a50d1106252

Figure 6 Example of provenance information in DataCite API response

3.5 Tracking changes

A major change to allow this provenance information to be implemented was the introduction of audit logging, this tracks every change to DOI metadata requested by DataCite clients. This means we are generating new provenance information on a individual basis for every request.

DataCite does not have historical data before the initial launch of this service, however we are gathering a significant amount of new data samples every day and this should provide new insight into the history of metadata going forward.

3.6 Result representation

DataCite decided to use PROV-DM and PROV-N, adapted to fit our existing JSONAPI representation format. While schema.org has the concept of actions in schema.org which provide similar functionality to activities, but they are not only about activities in the sense of PROV, but are more generally about activities on the web, e.g. the WatchAction for watching visual content on the web.

3.7 Initial feedback and next steps

The DataCite Activities API generates a lot of data, close to 10 million activity records in the first two months of operation. The service has been running very reliably during that time. So far there has been limited specific feedback by DataCite members, but the launch of the service has been appreciated them, e.g. at the DataCite General Assembly in April 2019.

At this point the activities API has not yet been integrated into the DOI Fabrica web frontend for managing DOI metadata. This will make this information more accessible to DataCite members. The API already allows reverting changes in DOI metadata, and this would be a useful and easy to implement feature in the DOI Fabrica web frontend.

With a rate of at least three million activities per month, the activities API will reach 50 million records by the end of the FREYA project, and it might then be useful to reconsider the architecture, e.g. limiting queries of the activities API to activities associated with a particular DOI.

4 Conclusions and future work

In this report we describe the work DataCite has done to implement an API tracking provenance for DOI metadata, as well as provenance work by ORCID and Crossref done in the last two years. The other PID services provider in FREYA, identifiers.org, does not store metadata for persistent identifiers and thus has a more limited need to track provenance, and their current conceptual work is described as well.

From discussions with their respective members, as well as discussion within the FREYA project, it became obvious that tracking metadata provenance is an essential component for providing trustworthy persistent identifier services. The new DataCite activities API focusses on tracking basic metadata provenance for DOIs, with use cases and focus on the ability to have an audit trail in the rare cases something goes wrong. The use cases described by ORCID and Crossref are more complex, as both the ORCID registry and Crossref/DataCite Event Data service allow contributions by multiple parties, making tracking provenance more complex. In the case of ORCID this also includes more advanced use cases, where additional organizations are involved in the provenance chain.

Besides these different use cases and technical implementations, the common theme in these implementations is that it is both desired functionality, but that the same time not a heavily used service. The situations where these services can be of value, e.g. when two claims in an ORCID record or the Event Data service are conflicting, are limited. Giving this context, it is important that provenance information for PID metadata is tracked, but that the implementation creates little overhead and doesn’t distract from developing and supporting services of more immediate value to users. Given the above, there is probably also little value in implementing provenance tracking in a standard way across all PID services providers, beyond relying on common concepts such as the PROV-DM data model.

DataCite used the basic PROV types and relations for its provenance implementation. ORCID did an extensive evaluation of the PROV ontology for their provenance implementation. The overall impression was that PROV provided value, but didn’t always meet the requirements, as it sometimes wasn’t a good match for what was needed for PID metadata provenance. Schema.org is a valuable metadata standard for scholarly outputs and used by several partners, but does not have good support for provenance, so that PROV, despite the shortcomings described above, is the best standard to describe provenance for PID metadata.

Going forward we don’t see much critical work left to do. DataCite needs to do more work on provenance in Event Data, coordinating with Crossref, and Crossref might want to also better track provenance of DOI metadata updates, building on the work done by DataCite in this deliverable. DataCite will include the DOI metadata provenance in its DOI Fabrica web frontend, and more work is needed to collect feedback from users regarding the need for provenance information.