THOR: Conceptual Model of Resolution

Document Information

Date: 13/09/2017

Authors: Martin Fenner (DataCite, orcid.org/0000-0003-1419-2405)
Robert Petryszak (EBI, orcid.org/0000-0001-6333-2182)
Rachael Kotarski (British Library, http://orcid.org/0000-0001-6843-7960)

Reviewers: Laurel Haak (ORCID)
Amir Aryani (ANDS)
Markus Stocker (PANGAEA)

Abstract: In this document, we look at three aspects of the resolution of identifiers to a URI representing the resource: dynamic data citation, content negotiation, and machine-enabled licence information.

We found the RDA best practices for dynamic data citation to be consistent with our recommendations for using DataCite DOIs for datasets, but neither fragment identifiers nor template handles, as alternative approaches, appropriate for this task.

Content negotiation resolves a resource, expressed as URI, to a number of different representations. We describe its status and open issues as implemented by THOR partners EMBL-EBI, ORCID and DataCite, and DataCite's refactored and relaunched DOI content negotiation service.

We discuss the need for normalisation of machine-readable licence information to enable rights-based query filtering, and the need for common file formats that combine content and metadata for datasets to enable large-scale adoption of direct machine-enabled access to data for re-use across information providers.

This work was supported by the THOR Project. The THOR project is funded by the European Union under H2020-EINFRA-2014-2 (Grant Agreement number 654039). The following report is based on a deliverable submitted to the European Union on 31 May 2017.

Visit http://project-thor.eu for more information.

Executive Summary

In this document, we describe work on one core aspect of persistent identifier (PID) functionality: resolution of the identifier to a URI representing the resource. The default behaviour is resolution to a landing page. With a focus on machine-enabled access, we look at three aspects of this functionality: dynamic data citation, content negotiation, and machine-enabled licence information.

The Research Data Alliance (RDA) Data Citation Working Group (WG) published best practices for dynamic data citation in 2015 (Rauber et al., 2015). We found that these best practices are consistent with our recommendations for using DataCite Digital Object Identifiers (DOIs) as persistent identifiers for datasets. We considered two alternative approaches to dynamic data citation, and found that neither fragment identifiers nor template handles were appropriate for this task. We will continue to work with the RDA Data Citation WG to help with the implementation of their recommendations.

Content negotiation is a standard protocol to resolve a resource, expressed as URI, to a number of different representations, specified by media type. In this report we describe the current status and open issues regarding content negotiation as implemented by THOR partners EMBL-EBI, ORCID and DataCite. As part of this work, DataCite refactored and relaunched its DOI content negotiation service to address the issues identified.

The main challenge with machine-readable licence information is the normalisation that enables filtering of queries based on the rights for re-use. For this it is important to express licence information as a URL and not a text string, and to normalise these URLs across information providers.

Through the work on conceptual models of resolution described above, we identified one important additional gap that needs to be addressed in order to facilitate the large-scale adoption of direct machine-enabled access to data: common file formats that combine content and metadata for datasets that are used beyond specific communities.

Contents

1 Introduction

Persistent identifiers (PIDs) for datasets and other scholarly works should be expressed as URLs and then, by default, resolve to landing pages with more information about the corresponding resources, including metadata and links to the content associated with these persistent identifiers (Starr et al., 2015). The landing page is one representation of the resource associated with the PID, and serves as a convenient entry point for human users.

Yet for machine-based access to resources using persistent identifiers, two important gaps exist in current implementations:

1. The relevant information provided on the landing page is not always machine-readable.
2. For automated workflows using machine-based access, direct access to the metadata or content is sometimes preferred, but existing implementations of these workflows impede the flow of information by accessing the landing page first.

One challenge in machine access to dataset content is the increased complexity compared to text-based documents. Text documents have a limited number of representations (for example, HTML, PDF and XML), plus multiple chapters or sections for larger documents such as books. For datasets, the following considerations need to be taken into account:

1. Multiple file formats, including common formats such as CSV, community specific file formats such as PDB (Protein Data Bank)[1], plus compression file formats such as ZIP for large datasets;
2. The need to retrieve subsets of a large dataset because the whole dataset is too large;
3. Evolving data, with changes happening every year, down to sub-second intervals;
4. Data that is merged from multiple data sources and then reprocessed, making it hard to keep track of provenance and licence information.

The goal of this report is not only to describe existing solutions, but to identify ways of automating workflows for accessing and citing datasets without the need to go via a landing page. Specifically, we look at data citation for evolving data, machine access to content directly from its persistent identifier, and machine-readable licence information for this content.

The output of this deliverable will feed into the implementation of these changes as part of the services delivered by THOR partners.

2 Implementing Dynamic Data Citation

One of the biggest challenges in data citation is the citation of evolving data – that is, datasets that are changing over time because data collection is ongoing. Evolving data can be versioned data. The THOR project has previously reported on the status and open issues of dataset versioning (Fenner et al., 2016b). Evolving data can also change in much more complex ways. This is frequently referred to as dynamic data.

[1] http://www.rcsb.org/pdb/static.do?p=file_formats/pdb/index.html

Three approaches currently exist for describing evolving data:

1. Fragment identifiers and query components
2. Template handles
3. RDA Data Citation WG recommendations: 'Data Citation of Evolving Data' (Rauber et al., 2015)

2.1 Query Components and Fragment Identifiers

'Fragment identifiers' and 'query components' are part of the URI specification[2,3], and have been widely used for web resources for more than twenty years. In the *URI Generic Syntax* (RFC3986)[4], they are defined as follows:

> *The start of the **query component** is indicated by the first question mark ("?") in a URI, and terminated by a hash sign ("#") or the end of the URI. The start of a **fragment identifier** is indicated by a hash sign ("#") and terminated by the end of the URI.*

Query components are processed by the server and fragment identifiers by the client, although this distinction was recently blurred in Javascript frontend frameworks such as Ember.js, where queries can also be processed by the client[5].

Query components complement other parameters such as userinfo, host, port and path in uniquely identifying a resource. The distinction between path and query component is not absolute, as path information can also be processed dynamically. A common practice is to use path information for a stable representation of a resource, and query components for optional parameters such as pagination or filters.

Query components do not work naturally with persistent identifiers expressed as a URI, as the syntax using a question mark is typically ignored or not allowed. The implementation of queries may also vary between different types of PIDs, as query strings have a special meaning in the Digital Object Identifier (DOI) proxy[6]. To implement queries in the DOI system, users should use handle templates, or register a DOI for a specific query (see below for both scenarios).

Fragment identifiers are interpreted by the client after the primary resource represented by the URI has been returned by the server. The client interprets the fragment identifier based on the media type. For example, it may use a different implementation for HTML and PDF[7]. A specification for the text/CSV media type exists[8]. Some media types are not a good fit for fragment identifiers, such as the image formats JPEG and PNG[9]. The XML-based SVG image format[10] is an alternative for using fragment identifiers with images.

---

[2] https://www.w3.org/TR/media-frags/
[3] https://tools.ietf.org/html/rfc3986
[4] Ibid.
[5] https://guides.emberjs.com/v2.12.0/routing/query-params/
[6] http://www.doi.org/factsheets/DOIProxy.html
[7] https://tools.ietf.org/html/rfc3778
[8] https://tools.ietf.org/html/rfc7111
[9] https://www.w3.org/TR/2009/WD-media-frags-reqs-20091217/
[10] https://www.w3.org/TR/SVG/linking.html

Fragment identifiers are typically used for navigation and bookmarking of supported media types, such as the start of a section in a HTML document, or the start and end of a video. As fragment identifiers are implemented in the client, no information is sent to the server about how fragments are being used and how often.

Since fragment identifiers are not sent to the server, they work with any URI, including a DOI or any other type of persistent identifier expressed as URI[11,12]. When using fragment identifiers with DOIs, it is important to remember that this use is outside of the DOI system, and implemented completely in the client. The DOI Handbook was recently updated to no longer describe fragment identifiers in the context of DOI names, as this confused some users who assumed special fragment identifier functionality in the DOI system[13].

2.2 Template Handles

In contrast to fragment identifiers that are part of the URI specification, template handles are a feature of the handle server software[14] used by handles and DOIs. Template handles overcome some of the limitations of fragment identifiers. Since they are implemented in the server, content is processed before being sent to the client. This allows for more flexible processing of data, which is useful when selecting a subset from a very large dataset.

Template handles were introduced to handle server software in 2010 (version 7.0)[15]. The main use case has been to provide globally unique persistent identifiers in the form of handles, without registering a record for each identifier. Adoption of template handles in the DOI community has been slow to take off; at present, we have not been able to identify any examples of dynamic data citation using template handles.

2.3 RDA Data Citation Recommendations

In October 2015, the Research Data Alliance (RDA) Data Citation Working Group published a set of recommendations for the 'Data Citation of Evolving Data' (Rauber et al., 2015). The working group is now in maintenance mode and is working with data centres implementing these recommendations. The basic assumption is that the same timestamped query must always return the same subset of the data, even several years and technology migrations later. The recommendations are listed below, followed by comments from the authors of this report; these comments have been shared with the RDA Data Citation Working Group.

R1. ***Data Versioning****: Apply versioning to ensure earlier states of data sets can be retrieved.*

R2. ***Timestamping****: Ensure that operations on data are timestamped, i.e. any additions, deletions are marked with a timestamp.*

R3. ***Query Store Facilities****: Provide means for storing queries and the associated metadata in order to re-execute them in the future.*

[11] https://www.doi.org/doi_handbook/5_Applications.html
[12] http://blog.martinfenner.org/2014/08/02/fragment-identifiers-and-dois/
[13] https://www.doi.org/doi_handbook/5_Applications.html
[14] http://www.handle.net/tech_manual/HN_Tech_Manual_8.pdf
[15] http://www.handle.net/HSj/hdl7_release_notes.html

R4. ***Query Uniqueness**: Re-write the query to a normalised form so that identical queries can be detected. Compute a checksum of the normalized query to efficiently detect identical queries.*

R5. ***Stable Sorting**: Ensure that the sorting of the records in the data set is unambiguous and reproducible*

R6. ***Result Set Verification**: Compute fixity information (checksum) of the query result set to enable verification of the correctness of a result upon re-execution.*

R7. ***Query Timestamping**: Assign a timestamp to the query based on the last update to the entire database (or the last update to the selection of data affected by the query or the query execution time). This allows retrieving the data as it existed at the time a user issued a query.*

R8. ***Query PID**: Assign a new PID to the query if either the query is new or if the result set returned from an earlier identical query is different due to changes in the data. Otherwise, return the existing PID.*

R9. ***Store Query**: Store query and metadata (e.g. PID, original and normalized query, query & result set checksum, timestamp, superset PID, data set description, and other) in the query store.*

R10. ***Automated Citation Texts**: Generate citation texts in the format prevalent in the designated community for lowering the barrier for citing the data. Include the PID into the citation text snippet.*

R11. ***Landing Page**: Make the PIDs resolve to a human readable landing page that provides the data (via query re-execution) and metadata, including a link to the superset (PID of the data source) and citation text snippet.*

R12. ***Machine Actionability**: Provide an API / machine actionable landing page to access metadata and data via query re-execution.*

R13. ***Technology Migration**: When data is migrated to a new representation (e.g. new database system, a new schema or a completely different technology), migrate also the queries and associated fixity information.*

R14. ***Migration Verification**: Verify successful data and query migration, ensuring that queries can be re-executed correctly.*

The recommendations are intended to allow the reproducible identification and citation of arbitrary views of data. To achieve this, the working group recommends the implementation of a dynamic, query-centric view of datasets. While the term **query** is often associated with relational databases, it is intended here as a more general concept that can also be applied to scenarios such as file-based data storage and version control systems.

Most of the recommendations then describe how this query store should be implemented. Requiring the same timestamped query to always return the same subset of the data implements recommendation #7 of the Joint Declaration of Data Citation Principles (JDDCP), 'Specificity and Verifiability' (Data Citation Synthesis Group, 2014). The concept of the query store, and the description of what needs to be considered when implementing this query store, should be considered best practices for any data centre, even when datasets are (relatively) static.

Although the RDA Working Group recommendations are intended to be generic, in some places it is clear that they were written with specific implementations in mind. One example is R5 (stable sorting). The more generic R6 (Result Set Verification) covers the principle to verify that the data returned by the same query should be identical, making R5 redundant.

The authors of the recommendations discourage the use of data exports/snapshots because they would come with more data storage overhead, and would not allow queries at arbitrary points in time. While these arguments are broadly valid, some data centres, such as those with slowly evolving data, may find data exports to be an appropriate strategy. So long as the principles described in the recommendations are followed, it should be left to the data centre to decide how to implement them.

The second part of the recommendations deals with issuing persistent identifiers for these queries. Rather than allowing the dynamic generation of queries by the user (for example, using template handles), it is recommended that a persistent identifier be issued for a query when needed, such as when a specific subset of data needs to be cited. These PIDs, together with metadata and other relevant information, are then stored internally, and persistent identifiers are issued using standard protocols. The one big difference to current practice is that the persistent identifier is issued when data are reused, not when they are generated. This approach puts the burden for enabling dynamic data citation on the data centre, with no major adjustments needed from the persistent identifier provider.

Minting persistent identifiers on demand also resolves another big challenge in data citation: databases holding very large numbers of datasets, such as those generated by sensors or sequencing machines, where only a small fraction will ever be cited or otherwise referenced by external systems. The overhead of issuing a persistent identifier for each dataset with standard metadata that is registered in a central index – the model for DOIs – is too high. Implementing a system similar to that recommended by the RDA Data Citation WG for evolving data would resolve this issue.

Some of the recommendations (R10, R11, R12) draw attention to best practices for data citation that have also been described elsewhere, for example by the Joint Declaration of Data Citation Principles (Data Citation Synthesis Group, 2014).

Although the recommendations are well written, they are intended primarily for a technical audience. Further outreach tailored to data centre managers and other less technical audiences could help with the adoption of these recommendations. One example for such outreach is a website explaining the recommendations in simpler terms[16].

2.4 Section Summary

The standard web approaches to dynamic content are query components and fragment identifiers, which form part of the URI specification and have been common practice on the web for a long time. Query components do not work well with persistent identifiers, as the query syntax using a question mark (“?”) is either not supported or is used differently by most persistent identifiers, including DOIs. Just as URI paths are frequently used to implement query component functionality, query component functionality can be implemented with persistent identifiers, for handles and DOIs by using handle templates.

Fragment identifiers are implemented in the client, such as a web browser, and therefore work with any persistent identifier expressed as URI. The implementation of fragment identifiers is specific to the media type. While media types such as text/html are well supported, fragment identifiers are not

[16] http://datacitation.eu/

appropriate for all media types (for example, JPEG or PNG), or a client implementation does not exist or is not widely adopted. The use of fragment identifiers assumes that the full resource is first returned by the server, which in many data citation use cases is not practical. Fragment identifiers are more limited than query components in terms of how a subset of a resource can be selected, and are typically limited to spatial or temporal information. It would therefore be impossible to implement the query store principles described by the RDA working group using fragment identifiers.

Fragment identifiers assume that the URI returns the resource itself, but it is best practice for scholarly content that the persistent identifier expressed as a HTTP URI redirects the user to a landing page with information about the resource (see Section 4: Content Negotiation). The use of fragment identifiers for datasets will typically require content negotiation, which complicates the implementation.

As fragment identifiers are implemented in the client, it is difficult – if not impossible – for the provider of the resource to track usage of fragments. This presents an obstacle, since many data centres want to collect this information.

Handle templates implement queries for handles and DOIs, and overcome most of the limitations of fragment identifiers described above. They return only the resource requested instead of requiring additional processing by the client, and have greater flexibility in how a subset of a resource can be selected. In contrast to fragment identifiers, handle templates are not widely used, and they have not seen much adoption since they were introduced in 2010.

The RDA recommendations for data citation of evolving data are also implemented around queries as the fundamental concept. The RDA working group has invested a lot of effort in describing work that needs to be done to make sure that the same query always returns exactly the same subset. As the resulting recommendations go into great detail about how queries should be implemented, it may appear as if the recommendations are hard to follow. In reality, this is simply the effort that is required to identify a subset of a dataset persistently. The same effort would be required using handle templates. The effort obviously depends on the size and complexity of the data. Moreover, although queries are a general concept, they are conceptually a more natural fit for data stored in a database compared to file-based solutions for data storage.

The fundamental difference between the RDA recommendations and handle templates is twofold:

1. Handle templates are implemented with the persistent identifier provider, whereas the RDA recommendations are implemented solely in the data centre.
2. Handle templates use a pattern to dynamically generate the identifier, whereas the RDA recommendations require a new persistent identifier to be minted every time a subset needs to be referenced.

The most notable advantage of handle templates is that they require less effort to implement. The main disadvantage, however, is that metadata for DOIs are stored outside the handle system, separately with each DOI registration agency. Many services provided by DataCite and other DOI registration agencies, including the tracking of data citations, rely on this additional information. For handle templates to work properly with DOIs, we therefore would need an implementation of handle templates in DOI registration services. No DOI registration agency is currently providing this integration.

The RDA recommendations can already be implemented today. A number of data centres have developed pilot implementations, or have already migrated their production systems[17]. As the functionality to provide persistent identifiers for queries is similar for all data centres, it makes sense that data centres using DOIs should take advantage of central services provided by DataCite and other DOI registration agencies. Handle templates could be part of that solution, but additional functionality would need to be built into the DOI registration system. Given that there has been no adoption of handle templates for DOIs, and that it is unclear as to how the management of DOI metadata (which traditionally has one record per DOI) works with handle templates, it remains to be seen whether handle templates will play a role in dynamic data citation going forward.

Registering a DOI when a resource is first used in a scholarly context, rather than when the resource is first made available, is a pattern that is not unique to dynamic data. Another common use case is the citation of web resources, such as blog posts, in the scholarly literature. We know that many of these cited resources will no longer be available a few years following publication (Klein et al., 2014). The preservation of these resources is an important challenge that the DOI registration agency Crossref is trying to address.

Fragment identifiers do not play a role in dynamic data citation, and it is unclear whether handle templates will play a role in the future. Going forward, DataCite will work with CNRI, the International DOI Foundation, Crossref, the RDA Data Citation WG and pilot data centres to implement a workflow that makes registration of DOIs for dynamic content based on queries as smooth as possible.

3 Content Negotiation

Content negotiation allows different representations of the same document to be served via the same URI. This is another important building block in our toolset that facilitates automated access to research data.

Persistent identifiers for datasets and other scholarly content should be globally unique and machine actionable (Data Citation Synthesis Group, 2014). The persistent identifier expressed as HTTP URI should redirect the user to a landing page for the scholarly resource that includes metadata and other relevant information (Fenner et al., 2016a). This landing page should also provide links through which the user can download the content, possibly in multiple formats. The landing page is thus the starting point for a human user who wants to access a dataset or other scholarly content.

The landing page also plays a central role for machine access to a scholarly resource, and should therefore include the relevant information provided to human users – metadata and links to the content itself – in a machine-readable format. Clear recommendations for dataset landing pages have been published, both as part of the THOR project and externally (Fenner et al., 2016b; Starr et al., 2015).

Content negotiation is an alternative approach that provides machine access to a scholarly resource using a persistent identifier. Instead of first going to a landing page, metadata or content itself can be

[17] https://www.rd-alliance.org/groups/data-citation-wg.html

accessed directly. Content negotiation is part of the HTTP standard[18], and can be used to provide access to different representations of the same resource using the same URL, but using different HTTP 'Accept' headers. Content negotiation is also used heavily to publish Linked Data[19], and is recommended in the 2011 Den Haag Persistent Identifier – Linked Open Data Manifesto[20].

The THOR partner EMBL-EBI is providing content negotiation both in its identifier.org resolving service[21], and directly from a number of databases[22]. At this point only content negotiation for **application/rdf+xml** is supported. For example, curl -L -H "Accept: application/rdf+xml" "http://rdf.ebi.ac.uk/resource/atlas/E-GEOD-14539" retrieves metadata for an Expression Atlas data in RDF/XML format.

3.1 DOI Content Negotiation

Content negotiation can also be used with DOIs expressed as HTTP/HTTPS URIs. The DOI registration agencies Crossref, DataCite, mEDRA and ISTIC all provide DOIs for scholarly content and, together with the International DOI Foundation (IDF), have been providing this service for many years[23]. A particularly popular feature is DOI content negotiation that returns a formatted citation in one of several thousand citation styles available via the Citation Style Language (CSL)[24]. DataCite provides a central citation formatting service at http://citation.crosscite.org/, available for DOIs from the DOI registration agencies mentioned above. Figure 1, for example, shows DataCite metadata in BibTeX format that was received via DOI content negotiation in DataCite Search.

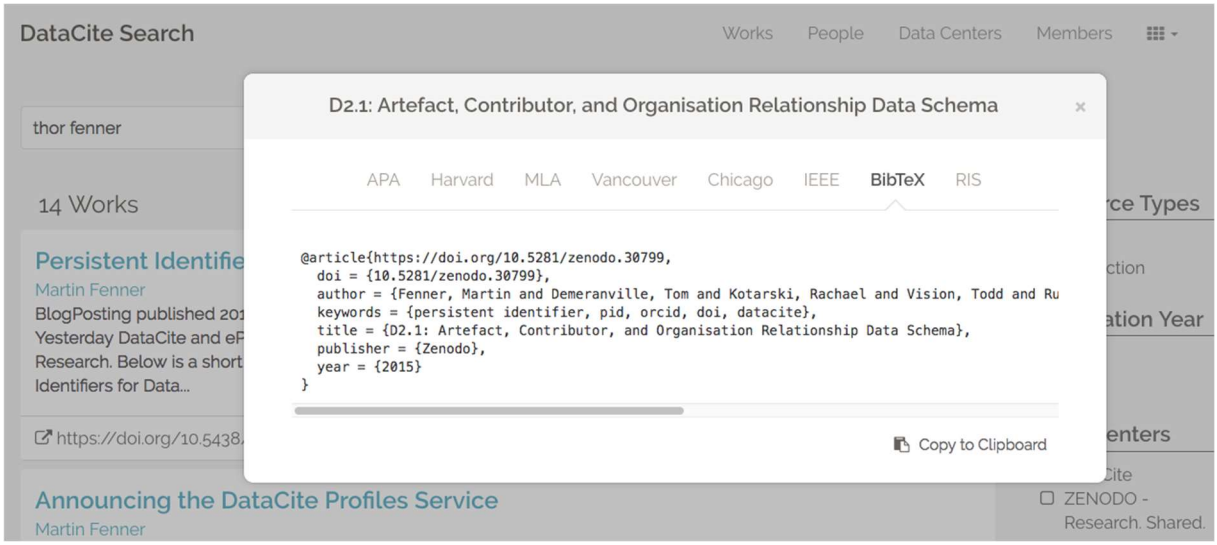

Figure 1: DataCite metadata in BibTeX format in DataCite Search, received via DOI content negotiation

[18] https://tools.ietf.org/html/rfc7231
[19] https://www.w3.org/TR/swbp-vocab-pub/
[20] http://ke-archive.stage.aerian.com/default.aspx%3Fid=462.html
[21] http://identifiers.org/documentation
[22] https://www.ebi.ac.uk/rdf/documentation/content-negotiation
[23] http://citation.crosscite.org/docs.html
[24] http://citationstyles.org/

As shown in Table 1, DOI content resolution can take one of four paths.

The main advantage of providing content negotiation via the DOI proxy service is that the same functionality can be provided for all DOIs, without the need for each data centre or publisher to implement this functionality in their own systems. Table 2 outlines the content types that are provided via the content negotiation services from Crossref, DataCite, and mEDRA.

Table 1: Possible actions in DOI content negotiation

Content Type	**Action**
text/html or */* (wildcard)	redirect to URL registered in the handle system
content type registered for DOI	redirect to content resolver of respective DOI registration agency, then redirect to URL registered for content type
content type known to content resolver, e.g. application/x-bibtex	redirect to content resolver of respective DOI registration agency, then return formatted metadata to user
unknown content type	redirect to content resolver of respective DOI registration agency, then return 406 Not Acceptable error

Table 2: Supported content types in DOI content negotiation[25]

Format	**Content Type**	**Crossref**	**DataCite**	**mEDRA**
RDF XML	application/rdf+xml	yes	yes	yes
RDF Turtle	text/turtle	yes	yes	yes
Citeproc JSON	application/vnd.citationstyles.csl+json	yes	yes	yes
Formatted text citation	text/x-bibliography	yes	yes	yes
RIS	application/x-research-info-systems	yes	yes	
BibTeX	application/x-bibtex	yes	yes	yes
Crossref Unixref XML	application/vnd.crossref.unixref+xml	yes		
DataCite XML	application/vnd.datacite.datacite+xml		yes	
Onyx for DOI	application/vnd.medra.onixdoi+xml			yes

[25] http://citation.crosscite.org/docs.html

The available content types fall into the following categories:

1. Metadata standards used by reference managers and other citation formatting tools (Citeproc JSON, RIS, BibTeX, formatted text citation)
2. Metadata standards of the respective DOI registration agency (Crossref, DataCite or mEDRA)
3. Metadata standards common across all DOI registration agencies (RDF/XML, RDF Turtle)

Formatting of citations is well served by DOI content negotiation, and widely used. Content negotiation is one mechanism by which users can obtain the full metadata registered with a particular DOI.

Users can register additional content types and the associated URLs for the DOIs they manage. This allows custom content types to be used with DOI content negotiation. An important use case is the registration of the content itself. For example, by registering 'text/csv' and/or 'application/vnd.ms-excel' and the associated URLs, content negotiation enables the resolution of a DOI directly to the URL for the content in CSV or Microsoft Excel format.

Content negotiation for custom types has not been used widely by the DataCite community, with only 26 data centres using this feature. Of those, the overwhelming majority of content available in this format are PDF files for documents, but there are examples of other content types:

1. **audio files**: https://data.datacite.org/10.17176/20170323-142432
2. **zipped data packages**: https://data.datacite.org/10.5284/1007940
3. **sequence data**: https://data.datacite.org/10.15156/BIO/SH026677.07FU

While the aim may be to allow automatic retrieval of content, this process only runs smoothly when no user interaction is required to access a file. Any data services where usage terms may need to be agreed to (for instance, providing access to a dataset including information about individuals) will present a problem for content negotiation to data files. For example, the small number of Archaeology Data Service files available for content negotiation require users to accept usage policy before proceeding.

In the context of a digitised British Library collection as a single dataset, content negotiation would allow a single DOI to refer to the dataset in terms of its images (application/pdf), the text content of those images as the output from optical character recognition (text/xml), and the metadata (application/json). In reality, the size of these files often requires them to be compressed. As the files will then be application/zip, it is not possible to use the current DOI content negotiation service. The current work around for this problem is to provide a separate identifier for each representation, yet this might not work for all organisations.

3.2 Limitations of the Current DOI Content Negotiation Implementation

While DOI content negotiation works well for formatting citations, or for obtaining metadata registered with a DOI, the current implementation of DOI content negotiation has important shortcomings:

1. There are no implementations across DOI registration agencies that represent metadata in a rich format that covers most of the metadata registered;

2. Some important metadata standards are not supported, such as Dublin Core[26] or DCAT[27];
3. Registered custom content types are not easily discoverable in a standard way;
4. Registering and use of custom content types, for example, for content, have poor adoption;
5. DOI content negotiation limits the content negotiation at the landing page.

While the RDF representations in XML and Turtle provided by all participating DOI registration agencies could be used to provide rich metadata for DOIs from multiple DOI registration agencies in one compatible format, the current RDF implementations have fairly limited metadata support. For example, the RDF in sample (1) is for a journal article that cites the dataset in sample (2):

1. https://api.crossref.org/works/10.7554/elife.01567/transform/application/rdf+xml
2. https://data.datacite.org/application/rdf+xml/10.5061/dryad.b835k

The RDF does not go beyond the citation metadata provided, for example, by Citeproc JSON. Furthermore, there is no article/data link in either RDF/XML representations.

The list of metadata standards supported by DOI content negotiation is relatively short. For example, it does not include common metadata standards such as Dublin Core, common metadata standards for data such as DCAT, or important community standards such as DDI[28] in the Social Sciences.

There is limited adoption of custom content types. Potential reasons are the limited promotion of content negotiation by DOI registration agencies, additional effort required during DOI registration, and the difficulty for the user to discover custom content types.

The current implementations of DOI content negotiation return a **406 Not Acceptable** HTTP status code when a content type is not recognised. One consequence of this is that it is not possible to offload content negotiation to happen at the landing page, for example, for custom content types.

3.3 Relaunch of the DataCite DOI Content Negotiation

To overcome the limitations described above, DataCite rewrote its DOI content negotiation service and launched it into production in May 2017 at https://data.datacite.org. The source code is available under an open licence (Fenner, 2017a).

The following considerations went into the refactored service:

1. Make the architecture generic, so that the service can be adapted to work with DOIs from multiple DOI registration agencies
2. Better support existing content types, such as page numbers, abstract and keywords in BibTeX, RIS and Citeproc JSON. Provide richer metadata for RDF/XML and Turtle
3. Support new content types, such as schema.org/JSON-LD
4. Forward the request to the landing page if the content type is not supported
5. Advertise available content types in the response header

[26] http://dublincore.org/

[27] https://www.w3.org/TR/vocab-dcat/

[28] https://www.ddialliance.org/

To provide a generic framework for DOI content negotiation, the new DataCite service consists of two parts:

1. **Bolognese**: a library for metadata conversion (Fenner 2017b)
2. **Content-Negotiation**: a web service as a wrapper for this library[30]

Bolognese is both a Ruby library and a command line tool, and can be installed with a single command. This enables local metadata conversions, and makes it easy to add support for additional content types or DOI registration agencies. Currently, DOIs from DataCite and Crossref are supported as input. The content negotiation web service is a Ruby on Rails API-only application that serves as a wrapper, such as caching, error handling, and integration of the DOI Citation Formatter service at http://citation.crosscite.org.

Bolognese has much better metadata support compared to the previous implementation of DataCite Content Negotiation. In some cases (RDF/XML, RDF Turtle, BibTeX) this is done by using existing libraries for these metadata formats[29,30,31], while in other cases much more effort went into manual mapping and conversion of metadata. Two areas, in particular, needed more work: a) mapping of resource types (as each metadata standard uses a different controlled list); and b) author name parsing. Mapping of resource types is often incomplete. DataCite, for example, uses resourceTypeGeneral "Text", but does not have a controlled vocabulary for the various text-based content types (journal article, conference proceeding, book chapter, and so on). Author name parsing is hard when a single field is used for people and organisations, and when there are no separate fields for given name and family name. To address the later point, givenName and familyName have been added as optional creator and contributor name properties in DataCite Metadata Schema 4.0 in 2016 (DataCite Metadata Working Group, 2016). Other limitations of DataCite metadata include the lack of properties for volume, issue and page numbers, but it could be argued that this kind of information is less relevant for digital content (Fenner, 2016c).

The relaunched DataCite Content Negotiation supports one important new content type: schema.org[32]. Schema.org was started in 2011 by Google, Microsoft, Yahoo and Yandex to promote schemas for structured data on the Internet, similar to how Dublin Core started in the 1990s. Metadata can be represented in three different encodings: RDFa, Microdata and JSON-LD. DataCite supports JSON-LD as the most appropriate format for content negotiation; the other two formats embed the metadata in HTML. DataCite metadata map well to schema.org, and we are in discussion with the schema.org community to resolve some of the open issues. The Force11 Data Citation Implementation Pilot (DCIP) for repositories recommended in 2016 that machine-readable metadata for datasets should be embedded in landing pages using schema.org metadata, and the work on the DOI content negotiation service facilitates this process.

DataCite Content Negotiation uses schema.org/JSON-LD as an intermediary step to generate RDF/XML and RDF Turtle. The advantage over mapping DataCite metadata directly to RDF (for example, using DataCite2RDF (Silvio Peroni et al., 2016)) is that the mapping can be consistent across DOI registration agencies, improving interoperability. Another advantage is that no separate DataCite-to-RDF mapping needs to be maintained, and resources can be focused on mapping DataCite metadata to schema.org.

[29] https://github.com/inukshuk/bibtex-ruby

[30] https://github.com/ruby-rdf/rdf-rdfxml

[31] https://github.com/ruby-rdf/rdf-turtle

[32] https://schema.org/

One other important change in the relaunched DataCite Content Negotiation is the handling of unknown content types. Instead of returning an error with status 406 Not Acceptable, the request will be forwarded to the URL registered in the handle system, allowing content negotiation to take place at the data centre or publisher. This provides more flexibility in how DOI resolution to content can be implemented: a) via DOI content negotiation, registering a content type/URL pair for a given DOI; or b) via content negotiation at the data centre. The former approach means that the data centre can offload content negotiation to a central service, whereas the latter approach provides more flexibility, such as using pattern matching to redirect to content.

Content itself often needs to be compressed because of its file size. If multiple content types need to be supported, and a content type that supports compression is not available, then an alternative to the generic "application/zip" is to provide HTTP compression (for example, using the gzip protocol).

The best place to advertise the available content types for a given DOI is in the HTTP header returned by the DOI proxy. This allows a machine to discover all available content types via a HTTP HEAD request. DataCite and Crossref are working with CNRI, which is running the handle service for all DOIs, to implement this functionality, described in more detail at signposting.org[33] and in van de Sompel (2015).

Finally, as part of the relaunch of the Content Negotiation Service, DataCite will promote the use of content negotiation to resolve a DOI directly to content. This includes raising awareness of the new option of content negotiation at the data centre, and working with data centres to remove existing hurdles to resolve a DOI to content directly, such as requiring cookies or click-through terms of use.

3.4 ORCID

Although not in the business of resolving identifiers to datasets, ORCID does implement content negotiation for machine-to-machine interaction. Types that are currently supported include RDF/XML, RDF Turtle and N3, based on feedback received in the ORCID and DataCite Interoperability Network (ODIN) project, the precursor to THOR. In addition, Citeproc CSL is supported on a per-work basis, enabling clients to request work information in a format that can be easily translated into a formatted citation.

ORCID is now investigating the use of the schema.org/JSON-LD format to better interoperate with DataCite and other identifier providers, with the intent of making the data provided via content negotiation of greater use to the community. Unifying around an approach to Linked Open Data and a specific metadata vocabulary will make it easier to build a full scale graph of persistent identifier use. ORCID hopes that this investigation will lead to new features that will better enable large scale graph processing services.

Linking ORCID with other persistent identifier services using a schema.org approach has been piloted by the Research Data Alliance DDRI WG[34] as part of the Research Graph[35] interoperability services (Wang et al., 2017). It has proven to be a stable and scalable way of representing the connections between people, places and things. One of the adopters of this service is Research Data Australia, the data discovery ser-

[33] http://signposting.org/publication_boundary/

[34] https://www.rd-alliance.org/groups/data-description-registry-interoperability.html

[35] http://researchgraph.org

vice of the Australian National Data Service (ANDS). ANDS is currently exploring the mapping between ORCID and schema.org, and regularly synchronises with ORCID, DataCite and other providers to keep the Research Data Australia connections up to date. ANDS has expressed interest in a native Person representation that would enable them to use a common representation, which can be easily exchanged and compared across infrastructures.

Providing a native schema.org Person mapping for ORCID presents challenges. Schema.org is centred around things rather than people, which means that while a CreativeWork or Dataset can have Contributors and Creators, it is harder to express the inverse relationship. Other links, such as affiliations, are represented in the ontology, so can be mapped with relative ease and standard properties such as identifiers and names are all present.

There are two ways to model Person−CreativeWork relationships in the current version of schema.org: use JSON-LD reverse properties[36], or a Role entity[37]. Each has its own drawbacks. Reversed properties can be difficult to interpret and can result in misleading representations. The Google parser will, for example, interpret an ORCID record as a set of CreativeWorks, each with one author rather than a Person with many CreativeWorks. The Role entity is the more accurate model, but increases the complexity of the representations and requires the client to understand the role values used.

The schema.org community is very active and open to making changes that provide demonstrable value. Coupled with the fact that much of the mapping work has been prototyped by ANDS and DataCite, it is anticipated that these challenges will be overcome in the near future.

3.5 Section Summary

Content negotiation is a core feature of many persistent identifiers[38], and is essential to facilitate machine-based access to metadata and the content itself. Content negotiation is often implemented only for RDF formats, as we have seen for EMBL-EBI databases and ORCID. Content negotiation for DOIs has a long history, and is particularly useful to provide common metadata representations independent of the DOI metadata schema used.

We have identified several shortcomings in the DataCite DOI content negotiation, and have refactored and relaunched the service to address these gaps. The refactored service also makes it easier to adapt the DOI content negotiation going forward. DataCite has also reached out to the other DOI registration agencies registering scholarly content for possible collaboration on further improving content negotiation. This includes ongoing work on adding the available content types to the HTTP header of the DOI proxy to facilitate the discovery of available content types.

[36] http://json-ld.org/spec/latest/json-ld/#reverse-properties

[37] http://blog.schema.org/2014/06/introducing-role.html

[38] http://ke-archive.stage.aerian.com/default.aspx%3Fid=462.html

4 Machine-Readable Licence Information

A standardised mechanism for the resolution of licence information to the data it controls, and vice versa, is needed to assist data archive users to both easily discover openly available data and understand what they are permitted to use it for. The mechanism must include an easily accessible and exhaustive list of licences controlling access to data held by a given institution, as well as licence-based filters that allow the user to further restrict results of their original query to match the expected usage scenario. Finally, each data record should contain easily identifiable licence information that controls access to that record. Representing licence information in a machine-readable format will further enhance semantic interpretation, and thus discoverability by search engines and third party data aggregators, and potentially decrease the likelihood of unintentional data misuse. Aside from licence information, terms of use are also often required, particularly if there are access restrictions for data containing sensitive or personal information.

4.1 EMBL-EBI Experience

EMBL-EBI makes its terms of use explicit[39], yet stops short of defining a default licence (it does not provide a URL either on the 'terms of use' page or in a machine readable form). Moreover, for any specific licencing information, it includes a disclaimer that defers the user to each individual resource. In the case of software, for example, the page states that it 'may be used by any individual for any purpose', but then directs the user to check if any 'specific exceptions are stated on the web page'. Similarly, in the case of data services, the terms-of-use page states that EMBL-EBI 'places no additional restrictions on the use or redistribution of the data available via its online services other than those provided by the original data owners.' Consequently, in all cases the user is tasked with locating the appropriate licence information 'somewhere on the EMBL-EBI website or associated source code repositories'. For historical reasons, perhaps, the number of different licences used across the EMBL-EBI resource is greater than one might expect (see Table 3).

Table 3: Licences used at EMBL-EBI

Symbol	Licence	Example
	references to terms of use document at http://www.ebi.ac.uk/about/terms-of-use	http://www.ebi.ac.uk/rdf/services/atlas/describe?uri=http%3A//rdf.ebi.ac.uk/dataset/atlas/13.07
CC-BY	Creative Commons Attribution	http://identifiers.org/
CC-BY-ND	Creative Commons Attribution-NoDerivs	http://www.ebi.ac.uk/ipd/licence.html http://www.uniprot.org/help/license
CC-BY-SA	Creative Commons Attribution-ShareAlike	http://www.ebi.ac.uk/training/online/glossary/ihop http://www.ebi.ac.uk/ena/data/view/PRJNA294707
CC0	CC Zero	http://www.ebi.ac.uk/ols/ontologies/pco http://pfam.xfam.org/about

[39] http://www.ebi.ac.uk/about/terms-of-use

For the user, this variety of licence types must be both confusing and hard to find. The licence type is rarely shown on a data record page, being mostly hidden somewhere in the 'about' section of each resource, and is all but excluded in machine readable form. One exception is UniProt, which includes the machine-readable (RDF) licence information in the source of http://www.uniprot.org, but also includes a comments section with a licence name and link to a licence page as part of each data record in txt format (for example, the CC section at http://www.uniprot.org/uniprot/P35582.txt).

The European Genome-Phenome Archive (EGA) of human data consented for biomedical research presents a different sort of challenge to the user. Each dataset is typically controlled by a different Data Access Committee (DAC), which formalises the access rights in its own Data Access Agreement. Each prospective EGA data user must apply to the DAC that owns the required dataset, and the application form states the rights and responsibilities with respect to that dataset for the applicant. Given that EGA acts solely as a custodian of the data on behalf of the various DACs, and the data access rights are out of EMBL-EBI control, the problem of systematising the access levels to EGA data is beyond the scope of the THOR project.

Convincing each resource to include its licencing information as part of each data record would most likely require extensive changes to data model, search index and web interfaces, and is thus unlikely to be achievable within the time constraints of the THOR project. EMBL-EBI has, however, achieved economies of scale in its search functionality by introducing an institute-wide search interface[40], powered by a Lucene index that is refreshed nightly from systematic data dumps by the majority of EMBL-EBI resources. In order to achieve EBI-wide data search by licence type as well as licence-based filters for search results, it is feasible that if each EMBL-EBI data dump provider adds a new licence field to each data record, containing a licence name and a URL, the EBI-wide search engine could easily index this information and provide it as filter facets on its search results page. It should also be relatively easy for each resource to include RDFa-formatted information on its home page and in each data record HTML page. We will explore the feasibility of a pilot project as part of THOR to illustrate how the above solution could work for an example EMBL-EBI resource.

4.2 DataCite Experience

Similar to EMBL-EBI, DataCite leaves the decision about licences up to individual data centres. This information can be included in DataCite metadata via the optional property **license** with attribute **licenseURL**. The licence information is prominently displayed in DataCite Search results, and can be used for queries of DataCite metadata (for example, to only return records for content using a CC0 waiver).

Queries are made easier with standard vocabularies for metadata properties. This is unfortunately not the case for licence information, as data centres can add any licence. The **licenseURL** attribute allows a certain degree of normalisation and is less vulnerable to differences in spelling than the **license** text field. Different text configurations, however, sometimes appear for the same licence.

Providing a dedicated search interface that includes licence information is on the DataCite roadmap. Once implemented, we can evaluate the extent to which this functionality is used, and whether it generates incentives for data centres to add licence information to DataCite metadata they provide.

[40] https://www.ebi.ac.uk/ebisearch/overview.ebi

4.3 Section Summary

Licence information is essential to enable the reuse of data. This includes the use of clear and widely adopted licences; the Creative Commons licences and CC0 licence waiver have now become de facto standards for scientific content, with a preference for CC0 for scientific data[41][42]. These licences should be displayed clearly in human-readable form, such as on the dataset landing page. Licence information should also be included in machine-readable form, ideally as part of the standard dataset metadata.

Terms of use are often needed in addition to licences, and the two should be clearly separated. Terms of use are important when access or reuse restrictions apply.

The licence URL is the easiest way to provide actionable licence information, and is also the easiest way to normalise licence information. SPDX[43] is an initiative in the software community to achieve these goals. Although some of the most important licences for datasets are also covered by SPDX, it has not seen much adoption in the data community. For the time being, licence URL is therefore the recommended mechanism to provide licence information in a standardised manner.

Going forward, THOR should work with data centres and organisations such as RDA to highlight the importance of providing licence information, and to increase the adoption of licence information in metadata.

5 Conclusions

In this report we described the current conceptual models of resolution of persistent identifiers for data to the underlying content. We focused our analysis on three important areas (dynamic data citation, content negotiation, and machine-readable licence information), and identified a number of important gaps. While we are on a good path, we have not reached the tipping point where machine access to datasets has become an effortless and prevalent activity.

The main gaps identified are as follows:

1. The most promising recommendation for the citation of evolving data has not yet been widely implemented, despite being endorsed by the Research Data Alliance (RDA). Conflicting approaches such as the use of fragment identifiers are still popular, even though they impose important limitations.
2. Resolution of persistent identifiers directly to content has not been widely implemented, despite existing technical implementations that have been available for many years.
3. Only a fraction of datasets come with machine-readable licence information, making it difficult to automate the process of reusing only datasets with appropriate licence information in an automated fashion.

[41] https://wiki.creativecommons.org/wiki/CC0_use_for_data

[42] https://wiki.creativecommons.org/wiki/CC0_FAQ#Does_CC0_require_others_who_use_my_work_to_give_me_attribution.3F

[43] https://spdx.org/

Machine-readable licence information, dynamic data citation and resolution directly to content are all important steps, but ultimately there is another gap further downstream that needs to be resolved:

4. Common file formats that combine data and metadata are a crucial factor for the wider reuse of datasets.

While the scholarly community is struggling with similar issues with text-based documents, PDF and XML are common formats to deliver content. While there is no widely adopted standard in the scholarly community to embed metadata in PDF documents, these standards exist for XML, such as JATS[44] in the life sciences and increasingly other disciplines, and TEI[45] in the digital humanities.

We need similar standards for datasets that encode both content and associated metadata. A large number of file formats specific to a particular database or set of databases exist, as do standards for metadata. Popular file formats for datasets, such as CSV, are poorly suited to include metadata that describe the dataset. DATS[46], a metadata standard for biomedical datasets, does not include the data itself, in contrast to JATS for text documents.

Going forward, two approaches are needed to increase the reuse of datasets:

1. Work with data centres to help with the implementation of the best practices described in this report, such as the citation of evolving data or content negotiation. DataCite, EMBL-EBI and the Research Data Alliance play an important role here.
2. Start conceptual work on a standard for the archiving and interchange of scientific data. This work should take into account existing work on encoding datasets in standard formats, and on related work done on text-based scholarly documents, in particular JATS.

This work should further facilitate the reuse of data, consistent with the FAIR Data Principles, which are designed to make data Findable, Accessible, Interoperable, and Re-usable[47].

[44] https://jats.nlm.nih.gov/
[45] http://www.tei-c.org/index.xml
[46] https://biocaddie.org/group/working-group/working-group-3-descriptive-metadata-datasets
[47] https://www.force11.org/group/fairgroup/fairprinciples

6 References

Data Citation Synthesis Group. (2014). Joint Declaration of Data Citation Principles. Martone M. (ed.) San Diego CA: FORCE11. Retrieved from https://www.force11.org/group/joint-declaration-data-citation-principles-final [accessed 25 April 2017]

DataCite Metadata Working Group. (2016). DataCite Metadata Schema Documentation for the Publication and Citation of Research Data v4.0. DataCite e.V. doi: doi.org/10.5438/0012

Fenner, Martin, Crosas, Mercè, Grethe, Jeffrey, Kennedy, David, Hermjakob, Henning, Rocca-Serra, Philippe, … Clark, Timothy. (2016a). A Data Citation Roadmap for Scholarly Data Repositories. doi: doi.org/10.1101/097196

Fenner, Martin, Demeranville, Tom, Kotarski, Rachael, Dasler, Robin, McEntyre, Johanna, de Mello, Guilherme, Vision, Todd, Dappert, Angela, and Adam Farquhar. (2016b). THOR: Conceptual Model of Persistent Identifier Linking. Zenodo. doi: doi.org/10.5281/ZENODO.48705

Fenner, Martin, 2016c. Mysteries in Reference Lists. *DataCite Blog*. Doi: doi.org/10.5438/CT8B-X1CE

Fenner, Martin. (2017a). Content-Negotiation: an API for DOI content negotiation. *DataCite Blog*. doi: doi.org/10.5438/t1jg-hvhn

Fenner, Martin. (2017b). Bolognese: a Ruby Library for Conversion of DOI Metadata. *DataCite Blog*. doi: doi.org/10.5438/n138-z3mk

Klein, Martin, Van de Sompel, Herbert, Sanderson, Robert, Shankar, Harihar, Balakireva, Lyudmila, Zhou, Ke, and Richard Tobin. (2014). Scholarly Context Not Found: One in Five Articles Suffers from Reference Rot. *PLoS ONE* 9(12), e115253. doi: doi.org/10.1371/journal.pone.0115253

Rauber, Andreas, Asmi, Ari, van Uytvanck, Dieter and Stefan Pröll. (2015). Data Citation of Evolving Data: Recommendations of the RDA Working Group on Data Citation (WGDC). *Research Data Alliance*. Retrieved from: https://www.rd-alliance.org/system/files/RDA-DC-Recommendations_151020.pdf [accessed 19 May 2017]

Rauber, Andreas, Asmi, Ari, van Uytvanck, Dieter and Stefan Proell. (2016). Data Citation of Evolving Data: Recommendations of the RDA Working Group on Data Citation (WGDC). *Research Data Alliance*. doi: doi.org/10.15497/RDA00016

Peroni, Silvio, Shotton, David, Ashton, Jan, Barton, Amy, Gramsbergen, Egbert, and Marie-Christine Jacquemot. (2016). DataCite2RDF: Mapping DataCite Metadata Schema 3.1 Terms to RDF [Dataset]. *Figshare*. doi: doi.org/10.6084/M9.FIGSHARE.2075356

Starr, Joan., Castro, Eleni, Crosas, Mercè, Dumontier, Michel, Downs, Robert R., Duerr, Ruth, … and Clark, Tim. (2015). Achieving Human and Machine Accessibility of Cited Data in Scholarly Publications. *PeerJ Computer Science* 1, e1. doi: doi.org/10.7717/peerj-cs.1

Van de Sompel, Herbert, and Michael L. Nelson. (2015). Reminiscing about 15 Years of Interoperability Efforts. *D-Lib Magazine* 21(11/12). doi: doi.org/10.1045/november2015-vandesompel

Wang, Jingbo, Aryani, Amir, Wyborn, Lesley, and Ben Evans. (2017). Providing Research Graph Data in JSON-LD using Schema.org. *The 4th WWW Workshop on Big Scholarly Data: Towards the Web of Scholars*. doi: doi.org/10.1145/3041021.3053052

Appendix A: Terminology

Additional terms are defined below:

Term	Definition
ANDS	Australian National Data Service. See http://www.ands.org.au/
API	Application programming interface
BibTeX	Tool and file format used to describe and process lists of references, mostly in conjunction with LaTeX documents. See http://www.bibtex.org/
CNRI	Corporation for National Research Initiatives
Crossref	Digital Object Identifier Registration Agency for scholarly publishing
CSL	Citation Style Language. See http://citationstyles.org/
DataCite	An organisation that develops and supports methods to locate, identify and cite data and other research objects. Specifically, DataCite develops and supports the standards behind persistent identifiers for data, and the members assign them. See https://www.datacite.org
DCAT	Data Catalog Vocabulary
DCIP	Data Citation Implementation Pilot
DOI	Digital Object Identifier
DoA	Description of Action
EC	European Commission
EMBL-EBI	European Bioinformatics Institute, part of the European Molecular Biology Laboratory
EU	European Union
EC-GA	Grant Agreement (including Annex I, the Description of Action) signed with the European Commission
GA	General Assembly
ID	Identifier
IDF	International DOI Foundation
ISTIC	International Science, Technology and Innovation Centre for South-South Cooperation
JDDCP	Joint Declaration of Data Citation Principles. See https://www.force11.org/group/joint-declaration-data-citation-principles-final
JSON	JavaScript Object Notation
mEDRA	Multilingual European Registration Agency of DOI, the standard persistent identifier for any form of intellectual property on a digital network.
ODIN	ORCID and DataCite Interoperability Network
ORCID	An organisation that creates and maintains a registry of unique researcher identifiers and a transparent method of linking research activities and outputs to these identifiers. See http://orcid.org
PDB	Protein Data Bank
PID	Persistent Identifier
RIS	Research Information System
RDA	Research Data Alliance. See https://www.rd-alliance.org/
SPDX	Software Package Data Exchange: standard format for communicating the components,

	licenses and copyrights associated with software packages.
TEI	Text Encoding Initiative
THOR	Technical and Human Infrastructure for Open Research. See https://project-thor.eu/
URI	Uniform Resource Identifier
WG	Working group

Appendix B: Project Summary

The **THOR** project establishes a sustainable international e-infrastructure for persistent identifiers that enables long-term access to critical information about the life cycle of research projects. It enables seamless integration between articles, data, and researcher information creating a wealth of open resources. This will result in reduced duplication, economies of scale, richer research services, and opportunities for innovation.

The project has four concrete aims:

1. Establishing interoperability
2. Integrating services
3. Building capacity
4. Achieving sustainability

The project will meet these aims by defining relations between contributors, research artefacts (including data), and organisations. We will incorporate these relationships into the ORCID and DataCite systems. We will also expand existing linkages between different types of identifiers and versions of artefacts to improve interoperability across platforms and integrate ORCID iDs into production systems for article and data submission services in pilot communities and beyond.

The consortium will develop systems to embed new PID resolution techniques into existing services to support seamless direct access to artefacts, and in particular data. We will create services to allow associations between datasets, articles, contributors and organisations at the time of submission. Building on these, we will deliver the means to integrate trans-disciplinary PID services in community-specific platforms, focussing on cross-linking, claiming mechanisms and data citation (guided by the FORCE 11 data citation principles[48]).

For more information, visit http://project-thor.eu or contact info@project-thor.eu

[48] https://www.force11.org/group/joint-declaration-data-citation-principles-final