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

D2.1 PID Resolution Services Best Practices

Deliverable type	Report
Dissemination level	Public
Due date	31 May 2018
Authors	Sarala Wimalaratne (EMBL-EBI) Martin Fenner (DataCite)
Abstract	This report describes approaches to PID resolution, and sets out best practices to be followed as well as future work in the area and a survey of the practices of different disciplines.
Status	Submitted to EC 25 June 2018

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

PID resolution is a core functionality of persistent identifiers This document introduces the types of resolution services, explains the differences between them the and current resolution practices employed by the FREYA partners providing PID infrastructure. Best practices in PID resolution are set out, with steps towards improving the existing resolution services during the FREYA project. The practices of a range of different disciplines are also surveyed.

Contents

1 Introduction

The central vision of the FREYA project is the PID Graph, connecting and integrating systems of persistent identifiers (PIDs), mapping relationships across a network of PIDs and serving as a basis for new services of value to communities and user groups. The PID Graph will build on and extend the existing PID infrastructure of established services, while allowing for incorporation of new services and indeed new PID types in future, through suitable governance mechanisms (the PID Commons).

FREYA's Work Package 2 is responsible for providing the core services that form the foundation of the PID Graph—advancing the capabilities of the services within FREYA, but also offering guidance for those outside the project on good practice in the context of the PID Graph. The focus of the work is on harmonizing the existing PID services infrastructure, and on facilitating the integration of PID core services into services built in FREYA's own pilot applications, within the European Open Science Cloud, and beyond.

PID resolution is a core functionality of persistent identifiers: indeed, in implementing the separation of identifier from location of a resource, it is absolutely central to the idea of PIDs. It enables interconnections between entities and resources identified by persistent identifiers, and thus plays an essential role in the construction of the PID Graph.

This document serves as information and guidance on how persistent identifier resolution services function and the ways they can be improved, highlighting various differences between resolver types, their relation to existing disciplinary use cases, and their overall advantages and disadvantages.

The intended audience for this document includes both repository managers and PID resolver service providers. For repository managers it shows how PID resolver services are essential for persistent access to scholarly resources. PID resolver providers and potential PID resolver providers will find guidance on what they should implement in their respective areas and a better understanding of existing practice and use cases.

The document is structured as an overview of the problems of reliably accessing scholarly resources. It describes how PID resolver services help address this problem, what basic and advanced functionalities PID resolver services need or could have, and lastly how PID resolver services are used in a different disciplinary contexts.

Subsequent work within FREYA on core services will cover metadata for PIDs, discovery services through a PID services registry, and common DOI search.

2 The need for PID resolution

Identifiers for scholarly resources should use “a persistent automated method for identification, globally unique, and widely used by a community”[1]. Automated, in this context, means that the PID resolution can be performed programmatically.

There is a large body of research that clearly shows that using HTTP URLs does not provide a reliable means to locate scholarly resources on the web, but rather that two major problems occur:

- Link rot: the resource is no longer is available under the given URL. Klein et al. found in their seminal work using more than one million publications, that one in five references using a URL were no longer accessible[2].
- Content drift: the content available under the original URL has changed. This might happen as often as in three out of four resources[3].

Link rot and content drift are problems for all resources made available on the web, but are a particular problem for scholarly resources, where the community expects that the scholarly record is preserved over much longer periods of time than the typical short-lived web content.

As time has proven, ensuring HTTP URLs resolve properly – by making them “cool URLs”[4] – is hard. To address the problems of link rot and content drift, the scholarly community has therefore taken a different approach and implemented persistent identifiers (PIDs) with their supporting infrastructure. Using persistent identifiers instead of HTTP URLs creates another abstraction layer that comes at a small cost (from the computational, infrastructure and administrative points of view) and should therefore be considered carefully, although its benefits largely outweigh its cost. Research data is a good example where not every dataset will be kept available for years, as a portion of research data will always be temporary, e.g. the vast amounts of raw data produced by instruments such as the CERN Large Hadron Collider, or temporary datasets produced in intermediary steps of data processing workflows. CERN Analysis preservation (see section 6.3) is an example of dealing with these kinds of research data.

One of the main function of PIDs is to decouple the resource identifier from the location of the resource. The PID is expressed as HTTP URL and users of that URL will be redirected to a different URL that represents the (or one of several possible) resource location. This decoupling opens the door for having multiple resource locations, increasing the system’s robustness, reliability and scalability. There are exceptions to this approach, e.g. PIDs for people that in the case of ORCID do not use redirection, but point to a central registry with information about that person.

This document describes PID resolution in more detail, including the different resolvers used and how they integrate, the basic functionalities needed for any PID resolver, and additional functionality that is not always implemented, but can add extra value.

We will describe two PID resolver services in more detail that together provide probably the majority of PID resolutions for scholarly resources:

- the Identifiers.org and N2T resolvers;

[1] Data Citation Synthesis Group. (2014). Joint Declaration of Data Citation Principles. https://doi.org/10.25490/a97f-egyk

[2] Klein, M., Sompel, H. V. de, Sanderson, R., Shankar, H., Balakireva, L., Zhou, K., & Tobin, R. (2014). Scholarly Context Not Found: One in Five Articles Suffers from Reference Rot. *PLOS ONE*, 9(12), e115253. https://doi.org/10.1371/journal.pone.0115253

[3] Jones, S. M., Sompel, H. V. de, Shankar, H., Klein, M., Tobin, R., & Grover, C. (2016). Scholarly Context Adrift: Three out of Four URI References Lead to Changed Content. *PLOS ONE*, 11(12), e0167475. http://dx.doi.org/10.1371/journal.pone.0167475

[4] https://www.w3.org/Provider/Style/URI

- the DOI resolver service based on the Handle System.

Our focus is on describing common functionalities, but also highlights implementation differences, driven by different use cases. We also make clear what functionalities these PID resolver services are currently missing, and what additional work the FREYA partners are planning to address those shortcomings.

3 Resolver services

3.1 Types of resolver services

Persistent identifiers (PIDs) and PID resolver services are essential for addressing the link rot and content drift problem. Resolving a persistent identifier to its landing page can involve different kinds of resolver services (Figure 1):

1. **Domain Name Service (DNS) resolver:** Resolves a hostname to an IP address.
2. **Local resolver**, e.g. load balancer, API gateway or web server: Redirects to a different host and/or path.
3. **Full resolver**, e.g. handle system: Redirects to a URL following a regular expression pattern, or a specific URL stored in the service.
4. **Meta-resolver**, e.g. identifiers.org or n2t: Redirects to a URL following a regular expression pattern.
5. **Single-service resolver:** some PIDs resolve to a single central resource, e.g. ORCID.

Each of these is considered in turn below.

DNS, local resolvers and single-service resolvers are commonly used for web resources, whereas meta-resolvers and full resolvers are specific services for scholarly resources.

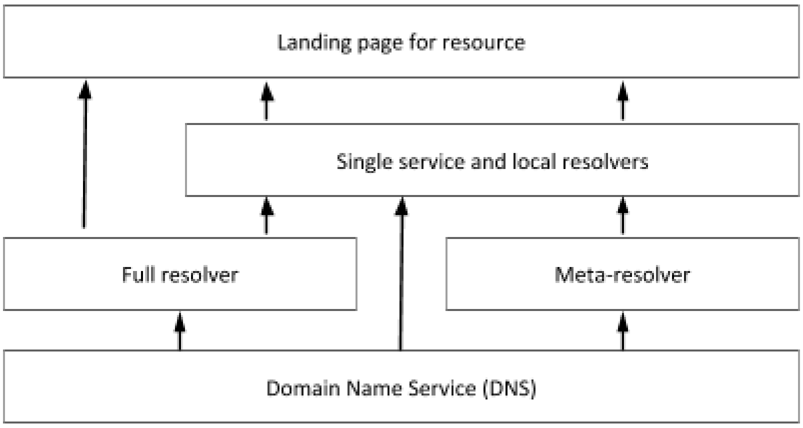
Landing page for resource
Single service and local resolvers
Full resolver
Meta-resolver
Domain Name Service (DNS)

Figure 1 Interaction between the different kinds of resolvers

While the handle resolver does not require DNS to operate, in practice the handle system is almost always accessed via a proxy that enables access via HTTP, e.g. https://handle.net or https://doi.org. Meta-resolvers depend on DNS. This means that DNS is always involved in PID resolution, and PIDs are best expressed as HTTP URLs to become machine identifiable.

Meta-resolvers in most cases require a local resolver to redirect to the landing page for the resource, as they use a pattern that includes the PID to construct the URL for redirection. The local resolver can then forward the request to the landing page, and the URL for that page may not include the PID.

Full resolvers can redirect to a landing page without the need of local resolvers, as they can store the specific URL for the landing page. In practice most URLs registered with a full resolver probably will still point to a local resolver. In the case of DataCite DOIs that seems to be true, but there is a small number of DOIs pointing directly to a landing page. DataCite have started to investigate the number and kind of redirections for URLs registered for their DOIs.

The responsibility for local resolvers sits with the data provider, whereas meta-resolvers, full resolvers and DNS are managed by external resources with input from the data provider.

3.2 Domain Name Service (DNS)

The web is made possible by the application of the Domain Name Service, a way to uniquely identify a resource and its location somewhere on the internet. This works by delegating authority over domains in a hierarchical fashion to various entities. The top levels are maintained by various organisations from government to non-profit and commercial, and then subdomains can be further assigned out, right down to individual ownership. This is in many ways analogous to what PID resolvers do, and PID resolver services leverage the DNS system and the web.

The significant differences are that with DNS, ownership is a problem when trying to ensure persistence, due to the potential individual ownership of domains. The burden of persistence lies with the owner, this can be a problem when domains change ownership or the authority in charge can no longer maintain. In addition, DNS provides a solution to resolve host names to IP addresses, but is not involved with resolving to a particular resource available at the host. DNS is thus an incomplete solution for PID resolution, but provides a central element that is complemented with other resolver services.

3.3 Local resolvers

When a content provider is handling the resolution at the local host level, this may be in the form of simple redirect lookups, API gateways, load balancers, or other configured sources to return a new location for a resource. Content providers usually handle some level of local resolution as this is fairly common within the web at large, it is generally good practice to always handle some level of resolution at the host level and depending on what content is being provided it is entirely suitable for the lookup to be handled at the local level. As this pattern is so common, Google and other indexers have come up with the concept of a canonical URL, which is the URL that should be preferably used when the same resource is available via multiple URLs via redirection.[5]

Local resolvers are an important element of any PID resolution strategy, but they can pose limitations when used without full resolvers or meta-resolvers:

- Local resolvers put the burden of implementing persistent URLs on each resource owner, instead of offloading most of the responsibility to a dedicated PID resolver service. This is particularly challenging when providing persistent links between many resources.
- Local resolvers depend solely on the resource owner to maintain the resolver (and DNS entry), not providing a fallback strategy when a resource is temporarily unavailable or no longer maintained.
- Local resolvers don’t usually have the economics of scale of a dedicated resolver service to provide advanced PID resolver services, e.g. link checking.

3.4 Full resolvers

A full resolution service for PIDs provides an entry point to access the content associated with a PID via a centralised authority. Generally for the context of the majority of PIDs, this resolution takes place on the web via a URL containing the PID, which then can redirect to the appropriate location of the content.

A full resolver can step in to solve the ownership problem by offering a centralised system, this can make the organisational aspects simpler due to the lower burden of responsibility on each provider of PID resources.

[5] https://support.google.com/webmasters/answer/139066?hl=en

The Handle System[6] is an example where PID resolution services have been built on top of various organisations to help maintain persistence. While handles can be used for a wide variety of use cases, at this point in time the handle system has seen the broadest adoption as PID resolver for scholarly resources.

The International DOI Foundation (IDF)[7] is a membership organization for members who want to provide persistent identifier services using the Handle System for persistent identifier resolution. FREYA consortium members Crossref and DataCite are IDF members and provide DOIs for scholarly resources. DOIs are handles using a specific namespace (10.x, e.g. 10.25490/a97f-egyk), and they provide additional functionality outside the handle system, most importantly standardized metadata and a commitment to persistence.

The handle system supports two ways of URL registration:

- Register a specific URL for a given PID, which is stored in the handle database, or
- Register a URL as regular expression pattern, where the PID becomes part of that URL (template handles).

Registration of a specific URL for a given PID (as for example all DOIs) requires more administrative and technical effort, but is needed for the following functionalities:

- can support resources that do not use a local resolver (and the registered URL doesn't follow a pattern);
- can systematically check all PIDs for link health;
- can store also core metadata for discovery services together with URL registration;
- allows "bounding" or enumerating the universe of identifiers. With template identifiers it is impossible to ask "do I have all of X?"

The handle system supports a number of advanced PID resolver functionalities (e.g. content negotiation), discussed in chapter 5.

The technical attributes of full resolvers are not their critical differentiating factor. Rather the most important characteristics of full resolvers and the associated identifiers are the norms and conventions for the stewardship of the identifiers and the things they point to. It is the community that defines the expected behaviour of the identifiers and regulates how the identifiers are managed. For example, they can require users of the full resolver system to:

- agree to a definition of "persistence";
- follow certain conventions for the assignment of identifiers (e.g when handling versions, relations, etc.);
- register metadata according to the community's standards;
- resolve identifier without the barrier of a paywall.

3.5 Meta-resolvers

Meta-resolvers are an additional layer of resolution that complement local resolvers, and provide a globally unique identifier via a unique namespace. In contrast to full resolvers, the meta-resolver does not require the registration of each persistent identifier, but only the registration of a unique prefix which corresponds to the assigning authority to clearly indicate all persistent identifiers provided by a specific resource. For example, the *Identifiers.org* meta-resolver maintain a list of resources and relevant metadata to support resolution[8]. Each resource is uniquely identified within the registry by assigning a prefix. The unique prefix combined with the local identifier forms the globally unique identifier which is used for resolution. The two

[6] http://www.cnri.reston.va.us/home/cstr/handle-overview.html
[7] https://www.doi.org/
[8] https://www.ebi.ac.uk/miriam/main/collections

meta-resolvers most widely used for scholarly PIDs are Identifiers.org[9] and N2T[10], and they are working together to use a common set of prefixes to resolve Compact Identifiers[11].

The meta-resolvers do not require registration of individual PIDs but relies on data providers registering and maintaining individual PIDs. This makes meta-resolvers particularly suited for already existing identifiers that are not yet globally unique, in particular accession numbers in the life sciences.

Since there is no PID registration, meta-resolvers cannot provide some of the advanced functionalities of PID resolution services, e.g. checking the availability of every PID single landing page, that are described in chapter 5. This means that currently, *identifiers.org* does not address the issues around link checking for an identifier that does not exist in a particular resource. Identifiers.org provides partial support for this by storing the identifier patterns, which allows the service to check whether a particular identifier is a valid pattern. However, this does not allow it to detect non-existing identifiers that meet the pattern specification.

One challenge for meta-resolvers is that they do not have the direct relationship with the resource provider that would allow them to define and enforce the same kinds of norms that the operator of a full resolver can. Even though, meta-resolver adds an extra level of dependency, it is necessary to maintain consistent access to the large number of diverse data repositories within life sciences where meta resolvers are largely in use.

Identifiers.org system provides a preliminary service for extracting schema.org metadata. The users can access this service at http://metadata.api.aws.identifiers.org/{compact_identifier}, e.g.: http://metadata.api.aws.identifiers.org/reactome:R-HSA-446203. Identifiers.org have tagged registry records with ‘schema.org’ tag to show the early adopters, https://www.ebi.ac.uk/miriam/main/tags/MIR:00600052.

The Identifiers.org registry supports content negotiation by recording resources that support content negotiation and using this information for redirection. For example: Uniprot RDF can be accessed via https://identifiers.org/uniprot/P12345.rdf.

Following the best practices detailed in this document (see section 2 and 3), improvements to Identifiers.org will be explored as part of the FREYA project. This will include:

- support for link checking using HTTP error codes and validity of the returned content to find out whether a particular link is valid;
- deploying the identifiers.org resolver system in different geographical locations;
- support for accessing and validating core metadata;
- support for content negotiation via HTTP header;
- support direct access to content via core metadata.

3.6 Single-service resolver

Single-service resolvers (in combination with DNS) are used by persistent identifiers that all point to the same resource. A good example is the ORCID identifier for contributors, which always redirects to the ORCID registry when expressed as HTTP URL. Since all ORCID IDs are resolved by a centralised PID resolution service, an additional layer of resolvers would not provide any additional relevant functionality.

[9] http://identifiers.org/

[10] http://n2t.net/

[11] Wimalaratne, S. M., Juty, N., Kunze, J., Janée, G., McMurry, J. A., Beard, N., … Clark, T. (2018). Uniform resolution of compact identifiers for biomedical data. *Scientific Data*, 5, 180029. https://doi.org/10.1038/sdata.2018.29

4 Best practices for basic PID resolver functionality

PID resolvers make persistent identifiers machine identifiable, a core PID functionality required for example in the Joint Declaration of Data Citation Principles (Data Citation Synthesis Group, 2014) and the Den Haag Manifesto[12].

As PID resolution works on a global scale, it is essential that all PIDs are globally unique and unambiguous, so that one identifier cannot point to a number of different resources. The ownership of a PID is on the Entity Body with the ability to issue PIDs of that type. In a global namespace, because of the nature of the current process of issuing PIDs by different organizations in the World, they are very likely to have collisions with other organizations' PIDs, thus, a mechanism is needed to avoid this problem, and this is done using prefixes (namespaces). A way to express the three dimensions of a globally unique ID (a resolution service, with prefix and PID information) is via a URL. For example:

https://identifiers.org/pdb:1A3N (resolver Identifiers.org, prefix pdb)

https://doi.org/10.1594/pangaea.887792 (resolver doi.org, prefix 10.1594)

A PID resolver thus needs to provide a list of prefixes, which should be unambiguous and thus maintained in a single place. The meta-resolvers Identifiers.org and N2T use a jointly managed prefix registry[13], where the prefixes are strings such as **pdb** in the example above. DOIs use handle prefixes in the format 10.x (e.g. **10.1594** in the example above), provided by CNRI[14]. At this point in time there is no single list of prefixes for all scholarly resources, but rather separate lists for the PID resolver services Identifiers.org and the Handle system. A mapping and possibly consolidation into a single prefix registry would increase the interoperability of PID resolver services. Table 1 shows examples of prefixes, identifiers and resulting meta-resolver URL from the Identifiers.org service:

Prefix	Identifier	compact identifiers	Meta-resolver URL
taxonomy	9606	taxonomy:9606	https://identifiers.org/taxonomy:9606
uniprot	P0DP23	uniprot:P0DP23	https://identifiers.org/uniprot:P0DP23
pdb	2gc4	pdb:2gc4	https://identifiers.org/pdb:2gc4

Table 1 Actionable compact identifier examples

For the administration of the PID resolver, basic management functionalities are required. Depending on whether or not only the prefix needs to be registered or each PID, the management functionalities need to be more or less complex. In the case of DOIs, this means that the registration of DOIs requires accounts with permissions, APIs and web user interfaces, documentation and support. Namespace management needs a similar list of functionalities, but is much smaller in scale (thousands of prefixes vs. millions of PIDs). In each case, basic status information about prefixes and PIDs should be provided, e.g. basic information such as date created and last updated, availability, and admin contact info.

A PID resolver should provide services that are reliable, performant, and with low latency. This is typically solved by running PID resolvers in multiple geographic locations, similar to the DNS system. This

[12] http://ke-archive.stage.aerian.com/default.aspx%3Fid=462.html
[13] https://github.com/identifiers-org/prefix
[14] https://www.handle.net/

redundancy can protect against service outages in particular physical locations. Latency can be a particular problem for users geographically far away from the nearest PID resolver, e.g. Australia if the service is only available in Europe.

As PID resolver services have become core dependencies of scholarly infrastructure, there is a need to always provide a current status of the service, as for example CNRI is doing for the DOI resolution service[15]. There is also a need for a fast response and status update if problems occur. Outages of PID resolver services, even if only for a few hours, can have a significant impact on the operation of our scholarly infrastructure[16].

Finally, PID resolver services should be sustainable: they should be a dependable part of our scholarly infrastructure, and not suddenly disappear or lapse in service quality. Bilder et al. have described the requirements for open scholarly infrastructure in their 2015 paper[17], and their recommendations can serve as a blueprint for organizations running PID resolver services.

[15] http://doi.statuspage.io/

[16] https://www.crossref.org/blog/january-2015-doi-outage-followup-report/

[17] Bilder, G., Lin, J., & Neylon, C. (2015). Principles for Open Scholarly Infrastructures-v1 [Data set]. Principles for Open Scholarly Infrastructures-v1. https://doi.org/10.6084/m9.figshare.1314859

5 Best practices for advanced PID resolver functionality

5.1 Advanced services

Advanced PID resolution services, usually provided by meta-resolvers, full resolvers or single-service resolvers, provide additional value to PID resolution. As centralized services they scale better than services implemented at each local resource, facilitating adoption. Several of these advanced services are work in progress, and not fully implemented.

5.2 Content negotiation

In general most resources expose some metadata about their data. These are commonly used for discovery, to provide provenance information, and to improve data interoperability. Some resources provide their metadata in a variety of metadata formats, and content negotiation is a standard approach provided by the HTTP protocol to retrieve these different formats from the same URL, according to the intended purpose. The *Den Haag Persistent Object Identifier –Linked Open Data Manifesto*[18] states as first principle "Make sure PIDs can be referred to as HTTP URIs, including support for content negotiation." Resolver services can support content negotiation, so that users are not redirected to the landing page for a resource, but instead receive metadata in a standard, machine-readable format.

Almost all resources expose their content in HTML format. Some data resources also provide the same content in other metadata formats to enable better machine accessibility. DataCite DOI content negotiation supports the following content types[19]:

Format	Name	Content Type	Supported
CrossRef Unixref XML	crossref	application/vnd.crossref.unixref+xml	Yes
DataCite XML	datacite	application/vnd.datacite.datacite+xml	Yes
DataCite JSON	datacite_json	application/vnd.datacite.datacite+json	Yes
Schema.org in JSON-LD	schema_org	application/vnd.schemaorg.ld+json	Yes
RDF XML	rdf_xml	application/rdf+xml	No
RDF Turtle	turtle	text/turtle	No
Citeproc JSON	citeproc	application/vnd.citationstyles.csl+json	Yes

[18] http://ke-archive.stage.aerian.com/default.aspx%3Fid=462.html
[19] https://support.datacite.org/docs/datacite-content-resolver

Codemeta	codemeta	application/vnd.codemeta.ld+json	Yes
JATS	jats	application/vnd.jats+xml	No
BibTeX	bibtex	application/x-bibtex	Yes
RIS	ris	application/x-research-info-systems	Yes
Crosscite	crosscite	application/vnd.crosscite.crosscite+json	Yes

Table 2 DataCite content negotiation option

The most widely used content negotiation format for DOIs is probably application/vnd.citationstyles.csl+ json, which is the input format for the DOI citation service[20], which is run by DataCite on behalf of several DOI registration agencies. The DOI citation formatter service provides a standard interface to provide a formatted citation in any of the more than 1,500 available citation styles[21] for any DOI from one of the participating DOI registration agencies (Crossref, DataCite, mEDRA, ISTIC).

ORCID supports content negotiation for ORCID URIs in addition to a specialised API. This provides metadata using the ORCID XML and JSON formats, as well as more general purpose schema.org JSON and RDF/turtle, N-triples and RDF/XML[22].

Content negotiation can be supported by any resolver type (full, meta or local resolver). Support by the meta-resolver or full resolver as in the case of DOI content negotiation requires access to metadata, but then provides a more standardized approach to content negotiation. Special content types that are used only in a particular community are probably better supported by local resolvers.

One of the challenges of content negotiation is the support of common metadata standards used by a large number of different resources. Dublin Core Metadata[23] are widely used, but do not always describe all relevant metadata, or describe them in a consistent manner. It is also a “flat” format, which makes it impossible to express relationships between metadata elements, e.g. which author name goes with which ORCID IDs. Citation formats such as BibTex and RIS only support a limited set of metadata relevant for generating a formatted citation.

Schema.org[24] is evolving as a metadata standard that can describe the relevant metadata for research data and other scholarly resources in enough detail. Schema.org is a collaborative initiative founded by the search providers Google, Bing, Yahoo and Yandex to mark up metadata about web pages. There is an active community behind schema.org and it is being widely adopted by other communities such as life sciences (Bioschemas[25]). Schema.org metadata are embedded into web pages (using for example JSON-LD and RDFa lite), and can be harvested using standard protocols.

DataCite is providing all its metadata in schema.org JSON-LD format via content negotiation and embedded into the search result pages of DataCite Search[26]. Data repositories have started to embed schema.org

[20] https://citation.crosscite.org
[21] https://github.com/citation-style-language/styles
[22] https://github.com/ORCID/ORCID-Source/blob/master/CONTENT_NEGOTIATION.md
[23] http://dublincore.org/
[24] https://schema.org/
[25] http://bioschemas.org/
[26] https://search.datacite.org/

metadata in dataset landing pages; a listing of about 30 implementations as of March 2018 is available[27]. ORCID has recently implemented schema.org JSON-LD for contributor metadata, including support for name variants, other person identifiers, works and affiliations[28].

5.3 Support for multiple locations

Resources can sometimes be made available in multiple locations, e.g. NCBI taxonomy IDs can be resolved to NCBI, OLS (at EMBL-EBI), etc. It is important that resolver services support the different decision trees about which location to resolve to. Relevant criteria include

- geographical;
- institutional;
- provenance;
- latency, preference;
- general technical availability.

The identifiers.org service supports redirection based on user preference and uptime. The handle system 10320/loc service supports redirection based on user preference, reputation (by giving a priority to specific locations), and geolocation. The 10320/loc service is not implemented for DataCite DOIs.

Regardless of any automatic decision making to which location should be returned for content, users often want the ability to decide for themselves. User preference should always be the first choice when provided, otherwise falling back to other considerations. Unless a user prefers a particular resource location, requests should normally be directed to the location with the highest reputation, e.g. the original location or most widely used location for a resource.

Response times can vary widely depending on a variety of factors and the users of resources can be spread across the globe, therefore a reasonable response time for content is something that is desired. Geographic location is often a good proxy for resource latency. Resources that can be reached reliably should be preferred over resources that are frequently unavailable. Service availability can be regularly checked by the resolver, and request redirection adjusted accordingly. Identifiers.org is using this approach.

The identifiers.org PID resolver supports linking of content to multiple locations, and this is a common practice in the life sciences. It does so by providing an additional layer of prefixes at the provider level (PROVIDER_CODE) to provide a simplified resolution to individual hosts. These codes are assigned at the institutional level or at the project level.

[27] Fenner, M., Crosas, M., Durand, G., Wimalaratne, S., Gräf, F., Hallett, R., … Clark, T. (2018). Listing Of Data Repositories That Embed Schema.Org Metadata In Dataset Landing Pages [Data set]. Zenodo. https://doi.org/10.5281/zenodo.1202174

[28] https://trello.com/c/PXT8Nw3o/4774-add-schemaorg-content-for-application-ldjson-requests-to-the-record-root-on-public-api

Provider	Provider code	Compact Identifier	Meta resolver URL
NCBI	ncbi	ncbi/taxonomy:9606	https://identifiers.org/ncbi/taxonomy:9606
EBI	ebi	ebi/taxonomy:9606	https://identifiers.org/ebi/taxonomy:9606
NCBO	bptl	bptl/taxonomy:9606	https://identifiers.org/bptl/taxonomy:9606

Table 3 Examples for provider codes

The Identifiers.org resolver provides a stable resolution service for these Compact Identifiers with provider codes (see Table 2). If user does not use a provider code for a collection that has multiple providers then the resolver selects a host by taking into consideration information such as the uptime and reliability of all available hosting resources.

5.4 Support for direct access to content

By default a persistent identifier resolves to a landing page for that resource that includes human readable and in some cases with machine readable information about the resource, and links to download the content. It can often be useful to access the content directly, bypassing the landing page, in particular for automated machine access. Unfortunately this functionality is currently for the most part not provided by PID resolver services and the resources they work with. One exception is Crossref, which allows publishers to register direct links to content in the metadata. Directly linking to content is a much bigger challenge for research data, with the variety of file formats, the granularity of access required for different use cases (e.g. only a small subset of a very large dataset) and sometimes very large file sizes.

The current best practice to have a persistent identifier point to a landing page, or provide metadata via content negotiation, means that the persistent identifier should probably never resolve to the content itself, directly or via content negotiation. Rather a separate URL for the content should be registered, as for example FREYA project partner Hindawi is doing in the metadata provided to Crossref for DOI https://doi.org/10.1002/cfg.114 (http://api.crossref.org/works/10.1002/cfg.114) :

```
"link": [{
  "URL": "http://downloads.hindawi.com/journals/ijg/2001/762302.pdf",
  "content-type": "application/pdf",
  "content-version": "vor",
  "intended-application": "text-mining"
}]
```

DataCite is providing a similar service that allows resource providers to register the URL and content-type for a given DOI. But this functionality is currently part of DOI content negotiation, which is confusing to users as access to metadata and access to content are mixed. The service is also not widely used. DataCite will rework the service implementation in 2018.

As many providers do not provide direct access to the raw data in a consistent method, meta-resolvers such as identifiers.org are unable to support this. However, if the providers are able to provide this information as metadata, for example using schema.org contentURL property embedded in landing pages, identifiers.org will be able to resolve to raw data.

Content in text format is usually provided in either XML or PDF format, with JATS[29] and TEI[30] being the most widely adopted XML standards. In contrast to text documents, such standard formats do not exist for research data, which might be provided in a variety of formats, including one or more CSV files, various media types within zip files or even tool specific formats related to the domain. This means that it can be difficult to obtain a direct link for PID providers to access, especially when there are multiple files involves.

A number of initiatives are working on solving this problem, and there was even a special session on this topic at the Research Data Alliance (RDA) Plenary in Berlin in March 2018: Approaches to Research Data Packaging[31]. Most initiatives in this space, which for example also includes BDBags[32], use the BagIt specification as a starting point. BagIt is an archive file format that can either include files or references to them, includes checksums for each file, and can have associated metadata that can help describe the content. Bagit is already in use by a variety of organisations and institutions and therefore can reasonably be pushed for adoption amongst data repositories. FREYA partners will work on a pilot implementation using the BagIt format.

One important use case for simplifying machine access to content is checks for content drift – the content available under the original URL has changed. Standardized machine access to content will make it much easier to implement services that monitor whether content has changed, using checksums, but also more complex approaches to comparing content.

5.5 Link checking

As resource content is available in such wide diverse distributed environments and can scale up into millions of distinct resource locations, it is challenging to ensure these resources are always available and do not change over time for a given PID pointing at them. This is important for the users of these resources, e.g. so that a citation of a dataset in a publication still points to exactly the same dataset three years later, when another researcher tries to reproduce the findings in the publication. Checking of this content can and is done by individual resource providers, however given the scale and diverse nature of this, any additional tools and services can make this simpler.

When approaching checking of contents and the links, there are various considerations that have to be taken into account, these can however vary between content providers:

- HTTP status codes;
- user agent;
- cookies and captchas;
- content type and landing pages;
- Javascript;
- metadata;
- “politeness”.

As most PID resolving services and the content they point to is accessible via the web, regular HTTP status codes can be leveraged to provide an initial view if the resource is available i.e. 200 OK vs an unavailable status i.e. 404 Not Found, 503 Service Unavailable, etc. One challenge when using status codes is that there are many services that incorrectly return a 200 OK status for unavailable resources accompanied by a human readable “not found” error message. In addition, HEAD requests are not routinely supported, requiring to fall back to more resource-intensive GET requests.

[29] https://jats.nlm.nih.gov/
[30] http://www.tei-c.org/index.xml
[31] https://rd-alliance.org/approaches-research-data-packaging-rda-11th-plenary-bof-meeting
[32] http://bd2k.ini.usc.edu/tools/bdbag/

Any automated crawler should always provider a user agent string, which contains information about the requestor, e.g. if it is an automated process, web browser, smartphone or other agent. Some content providers will still reject requests that include a user agent string, making automatic link checking via a central service more difficult. Cookies can sometimes be required to actually resolve the landing page, so steps need to be taken to ensure you store cookies, especially if the resolving goes through various redirects and/or specific handling maybe required if a known cookie has to be set. Content providers can sometimes mask behind a captcha specifically to stop automated crawlers, this obviously presents a challenge to any kind of automated link checking. In some cases, HEAD requests are honoured for browsers, but not other user agents, presumably to discourage automated access. In these situations specific conversations need to happen to allow automated checking, i.e. an authenticated API.

Content can be returned in numerous formats, this can vary from plain HTML to zips orPDFfs, depending on the content type. This can be a problem for content types other than HTML when trying to extract any information out of the format to determine how valid the link is.

HTML landing pages help with providing context for the content that is being provided, they also can help provide additional information when checking the availability and health. An examination of landing pages shows that the structure and layout is very different between content providers, especially across subject domains.

Sometimes the content can be generated by JavaScript. This is becoming even more prevalent in the age of SPA (Single Page Applications) and when doing any kind of automated link checking it has to be machine readable, otherwise using JavaScript capable automated tools i.e. a browser, are necessary to fetch and parse the content, slowing down and significantly complicating the process.

Metadata can sometimes be embedded within a page and is a good way to provide extra information to any link checking service. Metadata embedded as JSON-LD in schema.org format and/or as HTML meta tags in Dublin Core format are preferred.

When we are talking about link checking, we are making requests to the content provider, therefore it is important to consider being polite with any requests made. This is a topic that also needs to be handled with web crawlers in general.

In summary:

1. Encourage content providers to return accurate HTTP status codes that match the resulting content.
2. Use HTML landing pages to provide further context for content.
3. Use a specific user agent so link checking tools can be identified and white listed, and encourage providers to not block these agents.
4. Embed schema.org JSON+LD metadata in landing pages to describe the content.
5. When checking links avoid checks more frequency than every 10 seconds per domains.

DataCite has developed a prototype link checking service that sample checks across a variety of DOIs and plans to scale this up to be able to check all DOIs over time to help gather a broader picture of link health. For each provider Identifiers.org registry stores the access URL, example identifier and a keyword from the landing page. Using this information, identifiers.org has a link checking service that checks every resource provider in the registry every 24 hours. The result of this link checking service is stored in the database and used during the resolution time to make sure the user is served with a working link.

6 Disciplinary considerations

6.1 Life sciences

In life sciences, it is common practice for data repositories to assign local accession numbers to uniquely identify datasets locally. The accession numbers are generally made resolvable via local resolvers. In combination with meta-resolvers globally unique persistent identifiers are supported.

Some resources are deployed in multiple locations for high availability, but the main reason for the common practice in the life sciences to have several providers hosting the same data is additional functionality such as search, curation, visualisation features etc. This allows users to access the same information based on their individual needs.

Life science data repositories expose curated representations of the raw data. These curated representations are uniquely identified using local accession numbers. Resolvers provide access to the curated content. The raw data associated with the curated content is also usually available to download within the repository websites. The method of access generally depends on the repositories and the content they provide.

Life science data repositories provide access to metadata as a download, via APIs or content negotiation. Access to PID metadata via content negotiation is not common in the life sciences. The most widely provided content type is RDF. Accessing metadata via API is the preferred practice in the life sciences. Thus there is no consistent way of accessing the metadata. In addition, there is no consistency between the metadata being exposed making it difficult for a software agent to interpret the content. Bioschemas[33] is an effort to address these issues. The aim of the project is to encourage life sciences repositories to use schema.org markup to encode metadata on the landing pages to provide consistently structured information.

In terms of best practices for PID resolution, life science resources could benefit from:

- support for globally unique persistent identifiers via registering with a meta-resolver;
- encoding core metadata using schema.org metadata embedded into landing pages.

6.2 Earth and environmental sciences

Data centres that assign DOIs generally also use internal identifiers for their data sets. Some data centres like PANGAEA have a simple opaque mapping between the internal identifiers and the DOI name, so they can easily communicate the DOI name with the user before it is minted in the global handle system. Before making the data set public and minting a globally unique DOI name, the DOI name is using a local resolver (https://doi.pangaea.de), but after successful minting the global resolver (https://doi.org) is communicated with the scientist, e.g., https://doi.pangaea.de/10.1594/PANGAEA.881561 ⇒ https://doi.org/10.1594/PANGAEA.881561 Some datasets also refer to supplementary information (e.g., documentation) by using handles. Those are also globally unique. Otherwise, PANGAEA does not use any internal identifiers that are visible to the end-user.

In general, it is good to have multiple locations for storing the data sets, so a user can still access them if the repository is down (e.g., temporary network failure, slow download, ...). PANGAEA supports other institutions that keep copies of its datasets, but due to the fact that PANGAEA does not archive files, but instead uses a database backend to deliver the data to the user in multiple formats (that may change over time), it is not easy to create full and consistent copies. External data centres can only provide format specific views (e.g., CSV files) of PANGAEA's database contents. As those copies may differ from the original

[33] http://bioschemas.org/

database contents, PANGAEA will not allow to assign the same DOI name to them. Those copies are different representations of the same data. PANGAEA links those alternate representations in their metadata and the user can access them with different identifiers.

PANGAEA allows users to download the data files using several techniques next to providing the download link on the dataset landing page: Content negotiation in combination with machine-readable metadata approaches, schema.org JSON-LD and signposting[34]. Link checking is important to make sure that the user is not served with a dead link. PANGAEA returns HTTP status “404 Not Found” if a resource no longer exists. Schema.org JSON-LD metadata for PANGAEA dataset are available via the landing page and/or content negotiation.

6.3 High energy physics

Within CERN Analysis Preservation (CAP), a local UUID-styled local ID is used for identification purposes and to keep track of provenance information for the preserved analyses. CAP is based on the Invenio digital library framework[35], which has an internal UUID and a mapping table, so that way it is possible to connect the UUID to any outward-facing PID system in the future. Since CAP is a restricted-access platform, currently this ID is only used internally. The THOR project deliverable “Investigations into Extending Domain-Specific Implementations for PIDs” describes CERN’s considerations regarding the development of a local identifier system in detail.

In CERN Open Data, files can be downloaded directly from the resource landing page,, and data access is also possible via the XrootD protocol. Metadata is provided in a machine-readable format. A first prototype of a schema.org serializer that outputs in JSON-LD has been deployed.

In CAP, this information can be accessed via the cap-client, a client for interacting with our API. As mentioned above, every analysis in the system is identified with local, unique identifier. For actions such as listing, downloading or uploading files, It is necessary to have at least read access, as CAP is a closed system.

6.4 Humanities and social sciences

The British Library manages a wealth of content, itself identified by a variety of identifiers, ISSNs, ISBNs, DOIs, NBNs, URIs and DOIs. Due to legal deposit regulations not all copies of materials can be made available off site. This means that the British Library discovery systems, may have to route users to different versions and copies of items. Currently this is done using the SFX OpenURL resolver.

Users finding an article are presented with a number of options to accessing the content, and can select whether to access the digital item held by the library, via the publisher if the British Library holds a subscription or if it is free to access, or have a copy delivered to them via the British Library On Demand service. The drawback of using this knowledgebase-based approach to directing users to different copies is that it is only able to link users to the content at known access locations and using local identifiers. It does not make use of any persistent identifiers to identify alternative copies held elsewhere.

However, the British Library is investigating use of OADOI or Unpaywall to provide free access to items, which would sit alongside those examples. These links are already available for content not within the British Library catalogues, but development work will be needed to ensure that DOIs are included with all article records where one exists.

[34] Signposting the scholarly web, https://signposting.org/

[35] http://invenio-software.org/

7 Conclusions and recommendations

7.1 Conclusions

This document summarizes the current experiences and best practices for PID resolution services. It was written by organizations that provide a large part of the PID resolver infrastructure for the European Open Science Cloud and beyond. While there is no “one size fits all”, many best practices are implemented similarly across PID resolver services. We are also seeing a convergence of services and practices, and more of this will happen during the FREYA project as FREYA partners continue to work on these issues. PID resolver services are scholarly infrastructure, as such they are both essential and invisible unless there are problems. Infrastructure almost always benefits from standardization, as this keeps the cost down and reliably of services up, and PID resolver services are no exception.

The disciplinary perspectives of FREYA partners EMBL-EBI, PANGAEA, CERN and The British Library show that there are some disciplinary differences, but overall the needs and use cases are very similar across disciplines. We will look at the disciplinary implementation of PID resolution in much more detail in a separate report .

7.2 Remaining challenges

The biggest challenge remains adoption, using PIDs to identify, describe and locate scholarly resources. Adoption rates are high for scholarly literature, mixed for research data, and low for many other scholarly resources from scientific software to organizations. We will discuss persistent identifiers for new or evolving resource types in a separate FREYA report.

Lack of adoption does not simply mean that no persistent identifier exists, but could also mean that a local accession number or URL is used even when a PID exists. Much more outreach work is needed to increase adoption, and this report can help show the value provided by PID resolver services, and the problems that will occur when not using a PID resolver.

The main distinction between the two approaches to PID resolution described in this document (identifiers.org meta-resolver and doi.org full resolver) is PID registration. Registering every single PID enables functionalities that are otherwise impossible or difficult to support. On the other hand, PID registration requires extra work that might be too much for legacy resources created a long time ago, or for machine-generated data (e.g. sensors in the Earth and Environmental Sciences or Sequencing in the Life Sciences) produced in very large numbers. The two approaches to PID resolution can thus be seen as complementary, and we will continue to work on aligning them going forward.

PID registration is also something that can be provided in a number of ways. One interesting approach would use schema.org metadata embedded in landing pages, combined with sitemap files[36] to provide a complete listing of all resources available at this content provider. These metadata can be harvested and stored in a central service to provide some of the functionalities described in this report, e.g. link checking for all resources, and content negotiation (where schema.org metadata are converted into other metadata formats or a formatted citation).

7.3 Future work

This document has identified areas that need additional work, in particular in terms of aligning the two main PID resolver services for data described in this document, and in providing advanced PID resolver

[36] https://www.sitemaps.org/

functionality. The FREYA partners EMBL-EBI and DataCite will work on the following topics in the FREYA project going forward:

1. deployment of the Identifiers.org resolver in multiple geographic regions;
2. support for multiple locations per DOI in the DataCite DOI service;
3. harmonization of Identifiers.org and DataCite prefix registries;
4. implementation of a link checking service for all DataCite DOIs and larger samples of PIDs per resource in the Identifiers.org service;
5. support for accessing schema.org metadata in Identifiers.org;
6. better support for direct access to content in Identifiers.org and DataCite.

The work described above will further strengthen the production PID resolver services provided by DataCite and Identifiers.org, e.g. by link checking and redundant deployments in multiple regions. Mature and stable PID resolver services provided by FREYA partners are the foundation for the PID infrastructure in the European Open Science Cloud, used for example in disciplinary contexts from High-Energy Physics to Social Sciences.