

# Walking into the Future with PROV Pingback: An Application to OPeNDAP using Prizms

Timothy Lebo, Patrick West, and Deborah L. McGuinness

Tetherless World Constellation  
Rensselaer Polytechnic Institute  
Troy, NY, USA  
[lebot@rpi.edu](mailto:lebot@rpi.edu)  
<http://tw.rpi.edu>

**Abstract.** Adding provenance to existing systems can benefit users, but comes at an expense that may be difficult for some to justify. This trade-off can be overcome by *increasing* the value of provenance, by *decreasing* the cost to add it – or by doing both. This paper offers a contribution for each. First, we develop further the W3C PROV pingback technique so that it may reach its potential to interconnect provenance records that would traditionally sit in isolation, thus *increasing* their value. Second, we *reduce* the expense to publish the provenance of existing host systems by using minimal coupling to the Prizms Linked Data platform. Using an Earth Sciences scenario and the OPeNDAP data transport architecture as an example host system, we investigate how PROV pingback could work in practice, demonstrate its potential, and identify outstanding issues that must be addressed before it can be widely adopted.

**Keywords:** PROV, provenance, pingback, Linked Data, discovery

## 1 Introduction

The provenance community reached a significant milestone in 2013 when the World Wide Web Consortium (W3C) published its PROVenance documents. With a core model for provenance standardized, the community is now better prepared to turn their attention to subsequent challenges in research and application. In application, work may now focus on the relatively easier task of creating extensions that suit specific uses, which benefit from a common abstract structure and a growing set of interoperable tools. PROV was designed to suit Linked Data design principles [12], and publishing PROV as Linked Data offers great potential for distributed and uncoordinated discovery, access, and use of others' information. Conversely, PROV can benefit Linked Data by offering its consumers insight into how their distributedly-collected data came to be.

Unfortunately, the potential advantages of pairing PROV with Linked Data have yet to be seen at a scale as grand as the Web it uses. Now that PROV is a prominent fixture in the toolbox, a broader development community needs compelling reasons to adopt the W3C Recommendation and they need practical

answers for how to do it. Because existing host systems are often large and heavily invested in technologies not well suited to adopting Linked Data design to publish provenance records, solutions are needed to bridge the gap between existing systems and an interconnected Web of provenance with other systems. Our work aims to provide a technical foundation for such solutions, by developing PROV Pingback and applying designs from the Prizms platform.

PROV Pingback [9] has the potential to drastically interconnect provenance records that would traditionally sit in isolation. In contrast to the rest of PROV, which describes *how to describe* provenance so that anyone with the record may read about an object’s history, PROV Pingback enables parties to discover what happened to objects they created *after they have left their purview*. It addresses the practical need for upstream parties to obtain provenance recorded downstream, and does so with a simple technique based on the HTTP Link header.

The Prizms system emerged from the need to create high quality Linked Data [11] and has evolved into a Linked Data platform geared towards replicability, reproducibility, and transparency of the data that it publishes. Prizms supports the many Extract-Transform-Load processes that may be required to integrate a variety of others’ data about a topic of interest, and it provides for consistent provenance capture, metadata descriptions, and hosting using best practices.

The contribution of this paper is two-fold. First, it presents an approach to publish provenance of existing systems with very little effort; it allows them to expose provenance records without the overhead of publishing the records themselves and while benefiting from Linked Data principles. Second, this paper investigates the use of the PROV Pingback technique by applying it to a realistic scenario, demonstrating its potential, and identifying outstanding issues that need to be addressed before it can be mature enough for mainstream adoption. The work presented here can be used to both *increase* the value of provenance while *reducing* the effort required to add provenance to existing systems.

## 2 The State of the Linked PROV Cloud

Almost a year after standardization, PROV has not yet flourished within Linked Open Data (LOD). We present here two lightweight measures of PROV’s LOD presence using two resources popular within the Linked Data community: Open-Link Software’s LOD Cache and datahub.io’s dataset catalog. Attempts to provide a “State of the Linked PROV Cloud” suggest two challenges that the approach in this paper aims to address. First, it is possible that it is still too difficult for many to publish provenance in a manner that benefits a wider audience. Second, it is too difficult to discover existing provenance, even with Linked Data principles in place. Although widespread publication and discovery may not be a problem within individual applications (since first parties *receive* portions of provenance from which they can work), it remains an issue for those who wish to repurpose others’ existing data as an independent third party.

## 2.1 PROV Occurrences in OpenLink Software’s LOD Cache

OpenLink Software’s LOD Cache is a collection of 51 billion<sup>1</sup> RDF triples assembled over a period of years, and continues to grow as datasets come to the attention of its maintainers. We submitted SPARQL queries to find occurrences of the 50 classes and 68 properties in PROV. Table 1 shows the occurrences of the only fourteen PROV terms that occurred in the dataset. Most term’s occurrences are inconsequential, except perhaps `prov:wasDerivedFrom`’s 24 million (~12M from DBPedia pointing to Wikipedia pages and ~12M from wikidata.org). Unfortunately, these results do not portray a thriving PROV LOD ecosystem.

Table 1: Occurrences of PROV terms appearing in LOD Cache (20 Feb 2014).

|                                |            |
|--------------------------------|------------|
| Entity                         | 33         |
| <code>wasDerivedFrom</code>    | 24,975,410 |
| <code>hadPrimarySource</code>  | 7,874      |
| <code>generatedAtTime</code>   | 3,376      |
| <code>wasGeneratedBy</code>    | 33         |
| <code>wasAttributedTo</code>   | 33         |
| Activity                       | 214        |
| used                           | 214        |
| <code>startedAtTime</code>     | 214        |
| <code>wasAssociatedWith</code> | 214        |
| generated                      | 214        |
| <code>wasInformedBy</code>     | 106        |
| <code>endedAtTime</code>       | 108        |
| Agent                          | 1          |

## 2.2 PROV Occurrences in datahub.io’s Dataset Catalog

The datahub.io site should provide a more comprehensive and unbiased view of Linked Data, since anyone may contribute dataset listings. In addition to gathering entries for many other contemporary datasets, the site was used to organize the famous “LOD cloud diagram” between 2007 and 2011<sup>2</sup>, which established conventions for describing Linked Datasets within the CKAN data portal platform. According to the metadata at datahub.io<sup>3</sup>, fifteen datasets use the PROV vocabulary. Nine were created by the authors, so we set those aside. DBPedia is one, but we already saw it through the LOD Cache (above). That leaves five independent PROV adoptions (`imf-linked-data`, `bfs-linked-data`, `fao-linked-data`, `oecd-linked-data`, `ecb-linked-data`), but `imf-linked-data` can also be seen through the LOD Cache and all five were created by the same author and thus share similar structure. So, a community-based perspective on the use of PROV in LOD does not portray a thriving PROV LOD ecosystem, either.

<sup>1</sup> <http://lists.w3.org/Archives/Public/public-lod/2013May/0154.html>

<sup>2</sup> <http://lod-cloud.net>

<sup>3</sup> <http://datahub.io/dataset?tags=format-prov>

### 3 Approach

In this section, we describe our approach to easily create provenance leveraging the Prizms Linked Data platform, since it appears still too difficult to publish provenance according to Linked Data principles and it is still too difficult to discover provenance in LOD. First, we introduce the Prizms platform by creating datasets about the *structural* provenance of our example host system, OPeNDAP. OPeNDAP is a data transport architecture and protocol widely used by earth scientists to access remote data, such as satellite weather observations. We chose to use the OPeNDAP system to highlight how a system that *does not* use Linked Data principles can benefit from publishing its provenance records as Linked Data. Next, we describe how a minimal coupling to Prizms can publish a host system’s *behavioral* provenance, and discuss the distinction between *structural* and *behavioral* provenance. Then, we describe the addition of PROV Pingback to accept reports of downstream derivations of our host system’s data products. Finally, we demonstrate how the host can use its accumulation of clients’ provenance to easily lead others to those downstream derivations.

#### 3.1 Prizms’ “SDV” Dataset Organization: Source, Dataset, Version

We apply Prizms’ SDV organization principle throughout our approach. Prizms is a Linked Data platform designed to sustainably gather, integrate, and publish third party data to produce an integrated corpus about topics of interest. Prizms combines a few organizational principles, several existing toolsets, and commodity version control (Git) to facilitate coordination and collaboration among distributed team members. As a consequence, Prizms’ design facilitates within-team replicability and, by extension, reproducibility by external parties.

The SDV organization principle [11] organizes the many individual Extract-Transform-Load (ETL) processes that a data corpus or application may require according to three fundamental provenance aspects:

- *Source*, the agent (person, organization) providing the dataset.
- *Dataset*, a logical, abstract portion of the agent’s data.
- *Version*, a concrete portion of an agent’s abstract dataset.

Each of these three provenance aspects is identified using a concise identifier that follows a few conventions<sup>4</sup> (e.g. `usda-gov`, `national-nutrient-database`, and `release-26`) with the objective that a consumer could identify the original source agent, and the source agent could identify the dataset and version in their original holdings. The three aspects form a hierarchy for the datasets and serve as a naming scope for the entities mentioned within the datasets.

In the following example that we use to illustrate our approach, we establish six *abstract datasets* from three different *sources*. Because the datasets overlap in content but are created by drastically different means, it is important to organize them so that they can be properly managed. By following the SDV principle to organize provenance datasets, we are able to achieve provenance of provenance using the same mechanisms that are in place to express provenance of datasets.

---

<sup>4</sup> <https://github.com/timrdf/csv2rdf4lod-automation/wiki/SDV-organization>

### 3.2 A Concrete Basis: Modeling the Structure of the Host System

When a client requests a data product, its provenance often describes *behavioral* influences, such as the kinds of operations applied (e.g. filtering and aggregation), the mechanisms performing the operations, and their input data sources. It can be helpful, both from a designer’s perspective and from a user’s perspective, to supplement *behavioral* provenance with *structural* provenance. Structural provenance includes descriptions of the mechanisms performing the operations and how those mechanisms came to be. For example, software modules’ code repository changes are a rich source of their structural provenance. Provenance of an unfamiliar host system’s structure can help when designing the provenance of its behavior, since its components can be described *a priori* (e.g. modules’ versions, lifespans, and contributing developers) and can be directly referenced.

We described the structural provenance of OPeNDAP with three datasets. The first is a PROV-O representation of its Subversion (SVN) history<sup>5</sup>. The second is a curated list of software components along with their home in the code repository. The third connects the first two datasets by elaborating the SVN file path hierarchy. The following table shows the SDV aspects assigned to the structural provenance datasets, referred to in this paper as S1, S2, and S3.

|    | Source      | Dataset                    | Version     | Size  |
|----|-------------|----------------------------|-------------|-------|
| S1 | opendap-org | opendap                    | svn         | 1.9MT |
| S2 | us          | opendap-components         | 2014-Jan-07 | 1.4KT |
| S3 | us          | opendap-svn-file-hierarchy | 2014-Jan-20 | 1.0MT |

S1’s source agent is the OPeNDAP community; the dataset is the software itself, and its version is the latest SVN state. The repository’s XML log was transformed with XSLT to produce PROV-O<sup>6</sup>. S2 and S3 originated from the authors. S2 started as a spreadsheet and was transformed into Description of a Project<sup>7</sup> RDF using Prizms’ tabular converter. S3 was constructed by SPARQL querying for SVN file paths within S1/S2 and elaborating their hierarchy. These three datasets together describe the host system’s *structural* provenance and provided a basis for its *behavioral* provenance when handling data requests.

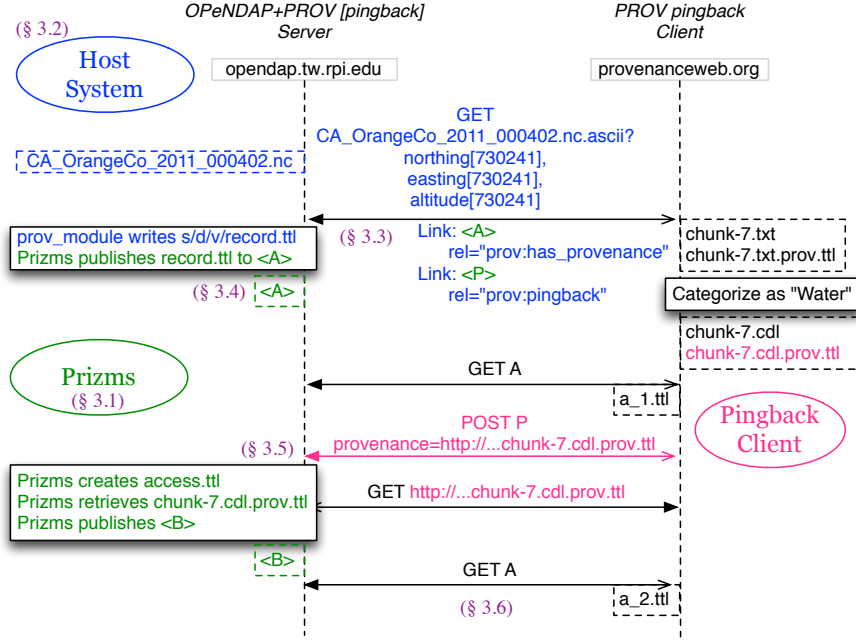
### 3.3 Minimal Modifications to the Host System (e.g. OPeNDAP)

While it remains the host system’s responsibility to record its own behavioral provenance (including references to its structural provenance), Prizms is used to reduce the effort required to publish those records as Linked Data. Figure 1 illustrates the coupling between Prizms and the host system, in relation to the downstream client that reports its derivations via PROV Pingback. In the upper left of the sequence diagram, a USGS LiDAR file `CA_OrangeCo_2011_000402.nc` is used by the host system to respond to the client’s HTTP request for `chunk-7`. While the host system processes the request as normal, it does only two additional things (§3.3, Fig. 1). First, it logs the provenance of its handling to a new

<sup>5</sup> The OPeNDAP source code is maintained at <https://scm.opendap.org/svn/>.

<sup>6</sup> Details at <https://github.com/timrdf/prizms/wiki/Publication:-IPAW-2014>.

<sup>7</sup> <https://github.com/edumbill/doap/wiki>



The minimal coupling between the host system (upper left) and Prizms (lower left), in relation to a pingback client (right). Section numbers indicate where each interaction is described in this paper.

Fig. 1: Sequence diagram among host system, Prizms, and pingback client.

file `s/d/v/record.ttl`. Second, it adds HTTP Link response headers pointing to `A` and `P` for the response's provenance and pingback, respectively. The host system required only five new parameters to coordinate with Prizms: Prizms' base URI (<http://opendap.tw.rpi.edu>), data directory root, and Pingback service URI (`/prov-pingback`), along with the SDV source and dataset identifiers for the dataset of provenance records (`us` and `opendap-prov`, respectively).

### 3.4 Prizms Publishes Host System's `prov:has_provenance` Target

Prizms' automation monitors for unpublished datasets to publish. The log file that the host system writes (e.g. `s/d/v/record.ttl`, above) triggers Prizms to publish it as Linked Data. The dataset URI `A` that results from writing the record in directory `s/d/v/` is the same URI that the host system returns in its `prov:has_provenance` Link header – this coordination is the extent of the coupling required for our approach. Although a custom publishing trigger was required to determine which records to publish in the dataset `us/opendap-prov`, it is available to be reused for other applications of our approach and employs the Vocabulary of Interlinked Data (VoID)<sup>8</sup> and PROV-O metadata that Prizms provides by default. A VoID Dataset `A` is named using its SDV aspects, its data dump is described and made available on the Web, and the provenance of loading

<sup>8</sup> <http://www.w3.org/TR/void/>

its dump file into a new SPARQL endpoint named graph is described. These best practices for publishing Linked Data facilitate its discovery and access.

### 3.5 Prizms Accepts Pingback Pointers

As shown to the right of Fig. 1, the client captures its own account of its request for a portion of the LiDAR file (e.g. in `chunk-7.txt.prov.ttl`). When making the HTTP request to the host system, the client must remember the pingback URI provided in the response header ( $P$ , Fig. 1) so that it knows where the host will accept reports of its derivations (see [9]). Once the client derives a product `chunk-7.cdl` from the host’s response, records provenance of its derivation in `chunk-7.cdl.prov.ttl`, and hosts it on the Web, the client can then report its results back to the host by accessing the pingback URI  $P$ . If the client manually loads the pingback URI using a Web browser, the service provides a description about the original request and accepts the client’s URL for provenance about `chunk-7.txt`. The service also describes to the user how the pingback may be performed automatically via HTTP POST using the *curl* command.

Prizms’ automation, which is centered around the SDV principle, allowed for a minimal pingback service implementation; it required less than 200 lines of code and can serve as a basis for other applications. When any Data Catalog Vocabulary (DCAT)<sup>9</sup> access metadata is situated within Prizms’ data root, Prizms acts on it to retrieve, integrate, and publish it. So, the pingback service’s only responsibility is to accept the pingback pointer and write it as access metadata into the same data root that the host system used for dataset  $A$ , using different SDV aspects similar to those shown in the table below. Doing so creates a new dataset  $B$  which is a local copy of the provenance hosted by the client.

Unfortunately, because pingback pointers could be provided and hosted by anyone on the Web, we cannot blindly trust that their contents are not malicious (e.g. executable code). To ameliorate this problem, we use Prizms’ trigger and secondary dataset frameworks to delete any pingbacks whose contents are not RDF containing PROV assertions. A dataset  $C$  is created for each batch of filtering. The following table shows the SDV organization for the three datasets created by the server after a single “request, pingback” cycle. Dataset  $A$  (Fig. 1) contains the provenance recorded by the host system during the client’s original request. Dataset  $B$  (Fig. 1) contains the host’s copy of the provenance reported by the client via pingback. Dataset  $C$  is the host’s aggregate of all its copies of provenance reported by clients within a recent duration (e.g. daily).

|   | Source            | Dataset                | Version           |
|---|-------------------|------------------------|-------------------|
| A | us                | opendap-prov           | 20140206-1391     |
| B | provenanceweb-org | prov-pingback          | 20140206-1391-1e2 |
| C | us                | pr-aggregate-pingbacks | 2014-Mar-03       |

### 3.6 Walking into the Future

Fig. 2a shows part of the HTML view when navigating to dataset  $A$ , the host’s original record of the client’s request for `chunk-7`. Even though the client’s cat-

<sup>9</sup> <http://www.w3.org/TR/vocab-dcat/>

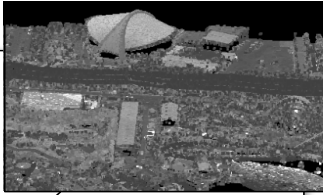
egorization and rendering (`chunk-7.cdl`, `CA_OrangeCo_2011_000402.png`) were created *after* this request, the host is still able to find and link to these derivations when describing the original request. Because Prizms accumulates the provenance pointed to by clients' pingbacks, it is able to use the single SPARQL query in Fig. 2b against only its own endpoint to find and offer links to client's subsequent derivations. The top portion of the query matches within the host system's account (dataset *A*, Fig. 1), and the bottom portion matches within the clients' (dataset *B*). The URL that the client requests (and that the host handles) is the natural link between accounts. With all of the relevant provenance in a single store and partitioned according to its source, the host is able to provide a variety of other Linked Data views to its clients. For example, the host can list all served requests with the files that they used, or the host can show the popularity of the files it serves based on the number of requests that used them or the number of downstream derivations that they contributed to.

**Supersets**

- [local\\_source\\_us\\_dataset:opendap-prov](#)

**Pingbacks**

- Local resource: [CA\\_OrangeCo\\_2011\\_000402.txt.cdl.nc](#)
  - Client's copy: [CA\\_OrangeCo\\_2011\\_000402.txt.cdl.nc](#)
    - Client's derivation: [CA\\_OrangeCo\\_2011\\_000402.png](#) (Portable Network Graphics)



(a) A portion of the HTML view of the `prov:has_provenance` dataset *A*, after a client has posted a PROV Pingback and Prizms has rehosted it as dataset *B*. The inset image shows a portion of the LiDAR rendering that the client derived.

```
select distinct ?host_input ?client_copy ?client_derivation ?format ?F
where {
  ?host_response
    foaf:isPrimaryTopicOf <A>;
    prov:wasDerivedFrom [ prov:specializationOf ?host_input ].

  ?host_input
    ^(prov:wasDerivedFrom | prov:wasQuotedFrom) ?client_copy.
  ?client_copy
    ^(prov:wasDerivedFrom | prov:wasQuotedFrom)+ ?client_derivation.
  optional { ?format ^dcterms:format ?client_derivation
    optional {?format dcterms:title ?F} }
}
```

(b) SPARQL query used by host to find downstream derivations of its data responses.

Fig. 2: Query and view of downstream derivations.



## 4 Discussion

**Related Work** Many methodologies exist for making systems provenance-aware. Of the dozen desiderata that Chapman and Jagadish [3] outline, our approach contributes to four: 1) building toward interoperability of provenance systems, 2) providing support for querying data and provenance together, 3) making provenance available to the user, and 4) capturing provenance of non-automated processes. PrIme [13] provides a step-by-step guide that we used in part to address the question “*What derivations have others made of this given data entity?*“. Because our approach does not address *what* a host system should record of its behavior, a methodology such as PrIme can be used to address such challenges. Groth et al. [4] present a technology-independent architecture of provenance systems, and discuss many valuable design considerations. Our low coupling approach follows their *SeparateStore* and *ContextPassing* patterns, yet after aggregating pingbacks it behaves similar to their *SharedStore* pattern.

Previous work has investigated Linked Data and provenance. Carroll et al. [2] established the central concept of a named graph. The concept has since been used by others [14], if only to capture provenance implicitly. The provenance recorded by our Prizms system employs the VoID and SPARQL Service Description vocabularies to describe named graphs as first class PROV entities. Hartig [6] distinguishes between *recordable* vs. *reliant* provenance on the Web. While the former is recorded by systems that can directly monitor their executions, the latter is accessed from third parties and requires evaluation to be trusted. PROV Pingback depends on (and benefits from) the combination of these two kinds of provenance and adds another means by which to obtain provenance from the Web (Hartig suggests DNS WHOIS, semantic sitemaps, POWDER, and Web service descriptions). Similar to our findings, he also concludes that “*there is only very little provenance-related, RDF-based metadata available on the Web*“ and points to lack of vocabularies, tools, and community sensitization/motivation as possible reasons. In follow on work, Hartig and Zhao [7] attempt to overcome the problem of missing provenance about Linked Data by offering a provenance vocabulary and extending several Linked Data publishing tools to automatically provide provenance. Instead of focusing on Linked Data provenance of Linked Data, we broadened the applicability of our Prizms provenance-aware Linked Data production platform by repurposing it to publish and interconnect provenance about non-Linked Data systems.

**Advantages and Limitations of our Approach** A key characteristic of our approach is the ability to frame PROV Pingback as a more fundamental dataset accumulation problem, thus reusing existing toolset’s automation, metadata, and provenance to achieve a qualitatively different kind of interconnectivity. SDV organization is a centerpiece of Prizms’ dataset accumulation, and stands as a design principle for systems that depend on many data sources. It can be seen as an answer to the request from Harth *et al.* [5] for a “social dimension” of Web provenance, so that data consumers can discuss

sources at a higher level of abstraction. They call for a formalism that could describe data placement policies for URI spaces. While SDV organization satisfied the need to identify socially-contextual sources and embeds source attribution within the design of entities' URIs (e.g., 300k, 1.1M, and 50 resources resources within `/source/pendap-org`, `/source/us`, and `/source/provenanceweb-org`, respectively), it similarly suffers from the DNS ambiguity that Harth *et al.* describe and would thus also benefit from a formalism for URI space ownership. Such a formalism could serve as a foundation for trusting those URI spaces and would have impact both when surveying Linked Data and when deciding if a pingback pointer is acceptable. The VoID vocabulary, with its `uriSpace` property<sup>10</sup>, might be a starting point for such a solution.

Our approach requires Linked Data design. While it may be considered a limitation by the host system, it allowed easy interconnection of distributed provenance systems with a simple RDF union. The dependency on HTTP Link also requires the host system to serve its data over HTTP. On the other hand, our approach allowed us to reuse existing vocabularies such as Friend of a Friend (FOAF) and existing instances such as DCTerms' file formats<sup>11</sup>. SPARQL 1.1 property paths also made it easy to traverse the many steps in a provenance graph to find all derivations. In our effort to gauge PROV's adoption in LOD, we considered several other sources that did not prove to be fruitful. Our objective was to find occurrences *in the wild*, *after* standardization, and discoverable using [semi-]automated means. Crawling all of Linked Data is the most comprehensive approach, but doing so is nontrivial [8]. A middle ground is for some to index Linked Data so that many others may perform centralized searches. The LOD Cache that we used is one example, but its manual, single-owner growth makes it a biased sample. Swoogle is a well-known index, but did not return any PROV terms. Sindice is a newer index that continues to accept pointers via a different pingback mechanism [10], but its accessibility has recently faded. Ping the Semantic Web, used in previous surveys [6], simply no longer exists. An alternative is to use a Linked Dataset catalog that anyone can contribute to. This has existed at <http://datahub.io/tag/lod> for seven years and is what we used as our second measure. In our view, this seems to be the best approach to discovering Linked Data sources. The Prizms system automatically provides the appropriate VoID descriptions and submits them to datahub.io on a weekly basis. Such a lightweight collection of pointers can facilitate more automated means to monitor and cache Linked Data sources. For example, Buil-Aranda *et al.* [1] currently monitor all SPARQL endpoints listed.

**Future Work** Despite its powerful ability to interconnect provenance records, PROV Pingback has a high potential for abuse (this is why our example service is not regularly available). Similar to many internet technologies, potential abuses need to be managed and can be mitigated through supporting infrastructure and tooling. Different applications should be able to control policies to adjust the

<sup>10</sup> <http://www.w3.org/TR/void/#pattern>

<sup>11</sup> <http://provenanceweb.org/instances/dcterms:FileFormat>

tradeoff between discoverability and abuses. Hosts can reduce their risk by being selective about which clients it offers pingback services to, based on information about the client or its request. A cautious pingback service should verify that every pingback submission is worthwhile, either by its URL (literally), URL contents, or by authenticating the client as a member of a trusted group. URL blocklists and whitelists can be helpful, but can become tedious to manage. URL contents should be handled with caution, perhaps to the point of performing it within a protected space and aborting it if it does not appear to be in an expected format. Any retrieved provenance should describe at least one derivation of a data product that the host served, otherwise it is not relevant. Authenticating the submitting client as a member of a trusted group could be achieved in a variety of ways, but one that does not require *a priori* coordination would allow for increased contributions and discoverability. Manual curation steps could also be used to validate any aspect used to determine worthwhile submissions.

A more complete and up-to-date *State of the Linked PROV Cloud* would serve as a design guide for provenance practitioners interested in adopting Linked Data principles, since it could verify that their published provenance is discoverable using traditional Linked Data means. Searches for terminology occurrences could be broadened by looking for non-PROV provenance terms or PROV extensions, accounting for reasoning, and by monitoring any dataset listed at datahub.io. Developers could use such a corpus to choose terms most appropriate for their application, based on quantitative measures of any term’s adoption.

We anticipate compounded advantages of a “Prizms network” when both clients and servers use the Prizms platform to propagate pingbacks. Techniques to combine PROV pingback with existing mechanisms such as Twitter’s “retweet” feature could accelerate community discovery of downstream derivations. Scalability of PROV Pingback should also be investigated, and simplifications of PROV Pingback could allow more direct usage by accepting the URI of the derivation itself and reusing the `prov:has_provenance` mechanism to find its provenance. Finally, the approach we presented should next be applied to *real* applications, not just *realistic*. In the case of LiDAR, we expect to apply it to a project with bathymetric and territorial data of New York State’s Lake George.

## 5 Conclusion

The symbiotic combination of PROV and Linked Data – both PROV *as* Linked Data and PROV *of* Linked Data – offers significant potential for distributed and uncoordinated discovery, access, and use of information. Unfortunately, these advantages have yet to be seen at a scale as grand as the Web it uses. Based on two lightweight measures that we present, it appears still too difficult or too unconvincing to publish provenance in a manner that benefits a wider audience.

We presented an approach to publish the *structural* and *behavioral* provenance of existing host systems by using minimal coupling to the Prizms platform, so that the host system’s provenance records may benefit as Linked Data even if its data cannot. We further described an implementation of the PROV Pingback

technique, demonstrated its potential to interconnect provenance records that would traditionally sit in isolation, and explored outstanding issues that need to be addressed before pingback can be widely adopted. By *decreasing* the cost to add provenance, and by *increasing* the value of provenance by forming an interconnected Web of provenance with other systems, the approach we describe can facilitate the adoption of provenance within a wider variety of applications.

## References

1. Carlos Buil-Aranda, Aidan Hogan, Jürgen Umbrich, and Pierre-Yves Vandenbussche. Sparql web-querying infrastructure: Ready for action? In Harith Alani et al., editor, *The Semantic Web – ISWC 2013*, volume 8219 of *Lecture Notes in Computer Science*, pages 277–293. Springer Berlin Heidelberg, 2013.
2. Jeremy J. Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. Named graphs, provenance and trust. In *Proceedings of the 14th International Conference on World Wide Web, WWW '05*, pages 613–622, New York, NY, USA, 2005. ACM.
3. Adriane Chapman and H. V. Jagadish. Issues in building practical provenance systems. *IEEE Data Eng. Bull.*, 30(4):38–43, 2007.
4. Paul Groth, Sheng Jiang, Simon Miles, Steve Munroe, Victor Tan, Sofia Tsasakou, and Luc Moreau. An architecture for provenance systems. Technical report, University of Southampton, February 2006.
5. Andreas Harth, Axel Polleres, and Stefan Decker. Towards a social provenance model for the web. 2007.
6. Olaf Hartig. Provenance information in the web of data. In *LDOW*, 2009.
7. Olaf Hartig and Jun Zhao. Publishing and consuming provenance metadata on the web of linked data. In *Provenance and Annotation of Data and Processes*, pages 78–90. Springer, 2010.
8. Aidan Hogan, Jürgen Umbrich, Andreas Harth, Richard Cyganiak, Axel Polleres, and Stefan Decker. An empirical survey of linked data conformance. *Web Semantics: Science, Services and Agents on the World Wide Web*, 14(0):14 – 44, 2012. Special Issue on Dealing with the Messiness of the Web of Data.
9. Graham Klyne, Paul Groth (eds.), Luc Moreau, Olaf Hartig, Yogesh Simmhan, James Myers, Timothy Lebo, Khalid Belhajjame, and Simon Miles. PROV-AQ: Provenance Access and Query. W3C Working Group Note NOTE-prov-aq-20130430, World Wide Web Consortium, April 2013.
10. Stuart Langridge and Ian Hickson. Pingback 1.0. Technical report, 2002.
11. Timothy Lebo, John S. Erickson, Li Ding, Alvaro Graves, Gregory Todd Williams, Dominic DiFranzo, Xian Li, James Michaelis, Jin Guang Zheng, Johanna Flores, Zhenning Shangguan, Deborah L. McGuinness, and Jim Hendler. *Producing and Using Linked Open Government Data in the TWC LOGD Portal*, pages 51–72. Springer New York, 2011.
12. Timothy Lebo, Satya Sahoo, Deborah McGuinness (eds.), Khalid Behajjame, James Cheney, David Corsar, Daniel Garijo, Stian Soiland-Reyes, Stephan Zednik, and Jun Zhao. PROV-O: The PROV Ontology. W3C Recommendation REC-prov-o-20130430, World Wide Web Consortium, October 2013.
13. Simon Miles, Paul Groth, Steve Munroe, and Luc Moreau. Prime: A methodology for developing provenance-aware applications. *ACM Trans. Softw. Eng. Methodol.*, 20(3):8:1–8:42, August 2011.
14. John Sheridan and Jeni Tennison. Linking uk government data. In *LDOW*, 2010.