Timestamped URLs as Persistent Identifiers

✉ Lars Gleim[1] and Stefan Decker[1,2]

[1] Chair of Information Systems, RWTH Aachen University, Aachen, Germany
{gleim,decker}@dbis.rwth-aachen.de
[2] Fraunhofer FIT, Sankt Augustin, Germany

Abstract. Uniform and persistent resource identifiers play a crucial role for sustainable data management and reuse in evolving knowledge graphs. Recent work identified explicit resource revisioning and immutability as important but insufficiently implemented and standardized components of corresponding data management systems. We propose the implementation of a global, persistent identifier system built upon time-based immutable resource revisioning of generic HTTP resources, as identified by their URL and resolved via time-based HTTP content negotiation, building upon existing Web standards. Supporting both distributed resource archival and state synchronization, the system would provide solutions to the problems of citation (referencing particular resource revisions), archiving (retrieving specific revisions), synchronization (change monitoring), and sustainability (preserving at scale, ensuring long-term access).

Keywords: Evolving Knowledge Graphs · Citation · Archiving · Synchronization · Sustainability · Persistent Identifiers · FactID · Dated URI · Memento Protocol

1 Data Management for Evolving Data on the Web

While the quantity of scientific, corporate, government and crowd-sourced data openly published on the Web grows, and data sharing, reuse, and integration become increasingly important for industry [5], the associated challenges of management and preservation of evolving resources and knowledge graphs remain insufficiently addressed to date [4]. The traditional view of digitally preserving datasets by "pickling and locking them away" for future use is fundamentally conflicting with the dynamic and continuous evolution of resources on the Web. Instead of archiving static bundles of resources at a certain point in time, dynamic data management solutions on the level of individual resources are needed to address the critical problems of (i) *citation* (how to cite a particular version of a resource), (ii) *archiving* (retrieving a specific version of a resource), (iii) *synchronization* (monitoring changes), and (iv) *sustainability* (preserving at scale, ensuring long-term access). Successful distributed version-control and collaborative systems show that data *immutability*, *referenceability*, *unique identification*, and *revisioning* are essential pillars to address these challenges [3]. The fundamental architecture of the Web further suggests that HTTP-based resolution and synchronization protocols should be employed for the resolution of corresponding actionable, global and *persistent resource identifiers* (PIDs).

While a large number of proposed approaches and standards are concerned with the fundamental problem of citation, especially the problems of archiving, synchronization, and sustainability remain insufficiently addressed to date.

In the following, we provide an overview of existing resource identification and resolution standards and their shortcomings w.r.t. evolving data on the Web. We then propose the implementation of a PID system reusing existing URLs as PIDs through the combination of dated URIs [10] with a resolution mechanism based on the HTTP Memento protocol [15], inspired by *FactID* [3]. We further argue how this system simultaneously addresses the citation, archiving, synchronization, and sustainability problem. Finally, we discuss open issues and the potential impact of the proposed approach.

2 Existing Identification Standards

Identifiers are generally divided into two categories; *handles* and *locators*. *Handles* are explicitly location-independent resource names that may be resolvable via the use of specialized resolution services and protocols, such as the Handle System [9] and its popular implementation *Digital Object Identifier* (DOI) [6]. In contrast, *locators* function like actionable pointers and enable direct resource dereferencing. The single most important locator of the Internet is the *Uniform Resource Locator* (URL) [1], core to the World Wide Web and principal foundation of resource-oriented architectures. In contrast to generic identifiers, PIDs are designed to enable particularly long-lasting, i.e., *persistent*, resource identification. Optimally, they should never be re-assigned, identifying the same identical resource eternally, and be resolvably indefinitely. Persistence is however purely a matter of service of the resource provider and not a property of any specific naming syntax [8].

While handle-based systems were traditionally described as advantageous in terms of identifier persistence and flexibility (since their resolution process may flexibly adapt to changes of the underlying resource and itself provide additional services such as functioning as a metadata store for the identified resources [16]) more recent accounts argue that the `http:` URI scheme can be employed to achieve similar features. [15,17] Well-known examples of such URL-based identification schemes include PURL [13] and indentifiers.org [7], which use HTTP redirection to resolve Persistent URLs into regular URLs according to the semantics of Cool URIs [12]. Even though virtually all identifier systems are effectively used and resolved via HTTP based dereferencing gateways nowadays [2,7], URL-based resolvers have been criticized since their inception [16]; mainly, for their dependence on the survival of their host domain name or since HTTP redirection does not allow multiple resolution (both limiting long-term sustainability).

To address this criticism, the *Archival Resource Key* (ARK) identifier system [8] introduced standardized and replaceable dereferencing gateways, combining aspects of both handles and locators. To identify a resource, a reusable handle, consisting of naming authority, resource name, and optional additional variant qualifier, is employed. A locator may then be derived from this handle by prefixing it with the hostname of an HTTP dereferencing gateway and persistence service for the resource identified by the handle. ARK thus enables the preservation of resource copies in multiple locations (i.e., different persistence services), providing a partial solution to the sustainability problem, as well as the management and retrieval of different versions of resources using plain HTTP-requests, providing a possible solution to the archiving problem. Nevertheless, ARK's general applicability to the Web of data is limited by its fixed URI scheme (significantly deviating from common URLs), its lack of synchronization support, inhomogeneous resource version management, and general lack of practical adoption.

All discussed approaches require the assignment, registration, and management of PIDs, typically distinct from and unrelated to already existing URL identifiers, creating the need for additional identifier mapping and discovery [15]. An identifier system for the holistic management of dynamically evolving data on the Web – providing compatibility with ubiquitous generic URLs, HTTP-resolution, and sustainable means for archiving and synchronization with minimal overhead – is missing to date.

3 A Persistent Identifier for Evolving Data on the Web

Instead of employing and managing single-purpose PIDs, we propose to employ existing URLs as PIDs through a simple resource revisioning mechanism, creating and exposing immutable, persistently identifiable revisions of arbitrary resources, as recently proposed by the abstract FactID scheme [3]. We propose a concrete implementation inspired by the ARK system, combining existing Web standards to create global, persistent resource identifiers from any URL using time-based content revisioning.

Citation. Revisiting the citation problem, the fundamental challenge persistent identifiers try to solve is that resources tend to change, move, or disappear over time. To address this issue, we employ the basic notion that at any given fixed point in time, any URL identifying a regular resource (or any locator for that matter) identifies only and exactly that resource. A simple persistent identifier can thus be created by combining an existing URL with a given point in time. That is also the underlying idea of Larry Masinter's 'duri' *dated URI* scheme [10]. A duri takes the form

```
duri:<timestamp>:<embeddedURI>
```

where <embeddedURI> is an absolute URI as defined in RFC3986 [1] and <timestamp> is a date-time, as per RFC3339 [11], allowing for the specification of an arbitrary time resolution. The meaning of such a duri is "the resource that was identified by the <embeddedURI> at the time given". [10] As such, duris provide a simple solution to the citation problem.

Archiving. To also address the archiving problem a corresponding resolution mechanism is required. A suitable candidate for duri resolution is the Memento protocol [14], which enables the retrieval of *Mementos* – historic states of resources – via time-based HTTP content-negotiation. The Memento framework distinguishes four logical components: Original Resource, Memento, TimeGate, and TimeMap. A Memento $\langle \mathtt{u}, \mathtt{t} \rangle$ captures the state of an Original Resource with URI u at a given point in time t (exposed via the Memento-Datetime HTTP header). Mementos are intended to be *immutable* and may optionally be associated with one or more distinct Memento URI(s) for *referenceability*. Such a URI-M must further identify the URL of its Original Resource in an HTTP Link header. Using the Accept-Datetime HTTP request header, historic states of Original Resources may then be requested from a so-called TimeGate through time-based content-negotiation, and are serviced through either a direct HTTP response

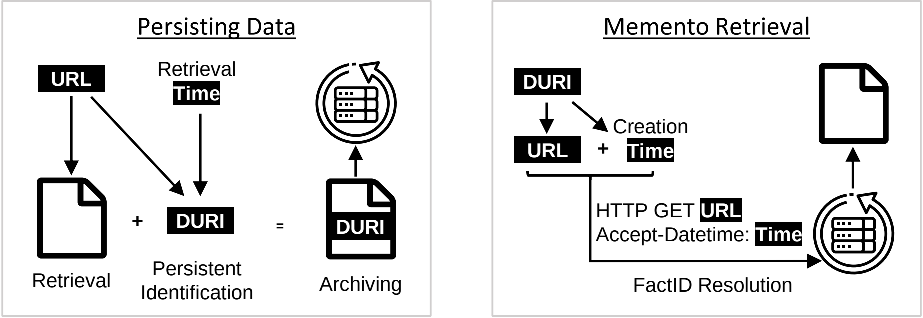

Fig. 1. Interaction of `duri` and Memento protocol when persisting or retrieving data.

or HTTP redirection to external archive locations, providing a simple solution to the archiving problem.

Synchronization. Additionally, the Memento protocol also enables revision discovery and synchronization through TimeMaps, which provide a listing of available Mementos $\langle \texttt{b}, \texttt{t}\star \rangle$ at points in time $\texttt{t}\star$ for a given Original Resource `b`, i.e., a history of available revisions with respective associated distinct Memento URIs. Thus, exposing up-to-date TimeMaps for resources enables trivial change monitoring and the discovery, retrieval, and thus synchronization of resource state.

Sustainability. Depending on the application scenario, TimeGate, TimeMap, and/or Mementos may all be provided by the original resource provider. It is however similarly possible to deploy all components independently of each other, as well as with optional redundancies. Notably, external TimeGates and Memento storage enable archiving to be conducted by third parties and on-demand, such as already provided by archive.org. Thus, adoption does not hinge on the support of any individual group or organization but may be adopted by interested users in backward compatibility with existing resources on the Web. Individual Mementos may further be resolved to multiple URI-Ms,i.e., different storage locations, (e.g. via TimeMaps), supporting explicit redundancy. As such, the Memento protocol also provides technological primitives to address the sustainability problem, by enabling flexible Memento resolution through one to many TimeGates and TimeMaps (similar to a handle), resolution of Mementos to multiple underlying URIs, and the ability to efficiently monitor resource changes.

Usage. Given the URL of an existing resource on the Web, a persistently referenceable and immutable version of it can be created by duplicating it to a Memento archive server and subsequently identifying it with a `duri` composed of its original URL and retrieval time, as illustrated in the left half of Figure 1. Given such a `duri`, the original resource state may then be retrieved from an archive through an HTTP GET request employing the Memento `Accept-Datetime` header with the Memento's creation time as specified in the `<timestamp>` part of the `duri`. The right half of Figure 1 illustrates the simplest case where the Original Resource serves as its own TimeGate. An overview of the additional available retrieval patterns is given in [14].

Open Challenges. While the combination of duri and Memento protocol already provides a good basis for the implementation of an interoperable data management system for evolving knowledge graphs and data on the Web, several limitations need to be addressed.

While the Memento protocol does allow for the retrieval of arbitrary data types, its timestamp format limits the *temporal resolution* of revisions to full seconds, clashing with the arbitrary temporal precision of RFC3339 timestamps employed by the duri scheme. Since many high-frequency applications, e.g., in the Internet of Things, require sub-second precision, we propose to adopt RFC3339 arbitrary resolution timestamps for time-based content negotiation in a revised version of the Memento protocol, as already employed in the duri scheme. In addition, future work should explore extending the protocol to *Memento creation*, not only their retrieval, to support further use case scenarios such as simple push-based state synchronization over HTTP, as well as providing HTTP REST APIs with the capability to uniquely identify created resource revisions using Memento headers in response to HTTP PUT and POST requests. Lastly, a promising extension of the duri scheme would be to allow for *empty timestamps*. The semantics of this would be to resolve the current state of the Original Resource directly whenever possible, or otherwise retrieve the latest Memento state available, enabling explicit referencing of the most recent revision of a resource.

4 Conclusion

In this paper, we proposed the implementation of a global, time-based, and resource-oriented PID system to address the sustainable management and preservation of evolving resources in knowledge graphs and on the Web. By combining existing URLs with timestamps through the duri URI scheme, we described a hybrid persistent identification scheme reusing URLs as handles. In conjunction with a resolution mechanism based upon the HTTP Memento protocol, such URL-based PIDs may serve as both locators and actionable handles (through a TimeGate) at the same time. Resource revisions may be physically stored anywhere, on a task-appropriate level of granularity and possibly duplicated, synchronized and archived in multiple locations in support of long-term sustainability. The proposed scheme further holds the potential to serve as a unified identification system for high precision time series data through to static concepts, as well as providing PIDs compatible with both generic binary data through to graph data formats such as RDF.

As such, the proposed approach provides a possible joint solution to the citation, archiving, synchronization, and sustainability problems, compatible with existing resource-oriented architectures and thus supporting interoperable data sharing, reuse, and integration for the increasing quantity of data published on the Web.

Acknowledgements. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612.

References

1. Berners-Lee, T., Fielding, R.T., Masinter, L.M.: Uniform Resource Identifier (URI): Generic Syntax. RFC 3986 (2005)
2. Crossref Display Guidelines (2017), https://doi.org/10.13003/5jchdy
3. Gleim, L., Pennekamp, J., et al.: FactDAG: Formalizing Data Interoperability in an Internet of Production. IEEE Internet of Things Journal pp. 1–1 (2020)
4. Gleim, L., Decker, S.: Open Challenges for the Management and Preservation of Evolving Data on the Web. In: MEPDaW @ ISWC (2020)
5. Ibarra, D., Ganzarain, J., Igartua, J.I.: Business model innovation through Industry 4.0: A review. Procedia Manufacturing **22**, 4–10 (2018)
6. ISO 26324:2012: Information and documentation — Digital object identifier system. Standard, International Organization for Standardization, Geneva, CH (2012)
7. Juty, N., Le Novere, N., Laibe, C.: Identifiers.org and MIRIAM Registry: community resources to provide persistent identification. Nucleic Acids Research **40**(D1), D580–D586 (2012)
8. Kunze, J.A.: Towards Electronic Persistence Using ARK Identifiers (2003)
9. Lannom, L., Boesch, B.P., Sun, S.: Handle System Overview. RFC 3650 (2003)
10. Masinter, L.M.: The 'tdb' and 'duri' URI schemes, based on dated URIs (2012), https://datatracker.ietf.org/doc/html/draft-masinter-dated-uri-10, work in Progress
11. Newman, C., Klyne, G.: Date and Time on the Internet: Timestamps. RFC 3339 (2002)
12. Sauermann, L., Cyganiak, R., Völkel, M.: Cool URIs for the Semantic Web (2007), http://www.w3.org/TR/cooluris/
13. Shafer, K.E., Weibel, S.L., Jul, E.: The PURL Project. Journal of Library Administration **34**(1-2), 123–125 (2001)
14. Van de Sompel, H., Nelson, M., Sanderson, R.: HTTP Framework for Time-Based Access to Resource States – Memento. RFC 7089 (2013)
15. Van de Sompel, H., Sanderson, R., Shankar, H., Klein, M.: Persistent Identifiers for Scholarly Assets and the Web: The Need for an Unambiguous Mapping. International Journal of Digital Curation **9**(1), 331–342 (2014)
16. Stone, L.: Competitive Evaluation of PURLs (2000), http://web.mit.edu/handle/www/purl-eval.html
17. Thompson, H.S., Orchard, D.: URNs, Namespaces and Registries (2006), https://www.w3.org/2001/tag/doc/URNsAndRegistries-50