IJDC | *Conference Paper*

Event Notifications and Event Logs: Transparent Sharing of Artifact Life Cycle Data

Patrick Hochstenbach
Ghent University

Ruben Verborgh
Ghent University - IMEC

Herbert Van de Sompel
DANS

Abstract

The “Event Notifications in Value-Adding Networks” specification provides an interoperable fabric that can be used in scholarly communication to exchange messages among data nodes that make scholarly artifacts available to the network and service nodes that add value to these artifacts. For example, a data repository can have a request-response conversation with a long-term archive that results in the latter relaying the coordinates of an archived version of the dataset to the repository. The push-oriented notification protocol is based on W3C Recommendations regarding the messaging protocol and payloads. Implementations of the protocol are in various stages of maturity, the most advanced being the COAR Notify effort that focuses on overlay peer review as a service. An important consequence, and actual design goal, of the conversational interoperability approach is the ability it provides to bi-directionally interlink the scholarly artifact and the service result in real-time, providing an attractive alternative to current interlinking approaches that by and large are heuristic-based and generate results with significant delays. Another consequence is the ability to publish an Event Log for each scholarly artifact that lists all the event notifications that were exchanged about it, providing full transparency about its entire life cycle, including where and how it was registered, archived, reviewed, and commented on. This paper describes essential aspects of the Event Notification protocol and illustrates it using a scenario. It then describes the Event Logs concept and illustrates it by means of that same scenario. It then gives an overview of challenges related to specifying Event Logs that are currently under investigation and largely relate to equipping them with affordances to make them verifiable and trustworthy.

Submitted 7 February 2024 ~ *Accepted* 22 February 2024

Correspondence should be addressed to Patrick Hochstenbach, Ghent University Library, Rozier 9, 9000 Ghent, Belgium. Email: Patrick.Hochstenbach@UGent.be

This paper was presented at the International Digital Curation Conference IDCC24, 19-21 February 2024

The *International Journal of Digital Curation* is an international journal committed to scholarly excellence and dedicated to the advancement of digital curation across a wide range of sectors. The IJDC is published by the University of Edinburgh on behalf of the Digital Curation Centre. ISSN: 1746-8256. URL: http://www.ijdc.net/

http://dx.doi.org/10.2218/ijdc.v19i1.920
DOI: 10.2218/ijdc.v19i1.920

Introduction

Motivated by the detrimental effects of the ongoing monopolization of research communication[1,2] (Larivière et al., 2015), the Andrew W. Mellon-funded ResearcherPod project set out in 2020 to investigate the technical feasibility of an alternative, decentralized communication system. The envisioned communication network aligns with COAR's Next Generation Repository (Bollini et al., 2017) perspective in which repositories, empowered by interoperability affordances, play an active role instead of merely being the final resting ground for scholarly artifacts. Although the ResearcherPod project pays special attention to the position of personal data pods as repositories in such a communication network, it remains realistic and actively considers general-purpose, institutional, and discipline-oriented repositories.

The envisioned network consists of nodes that are either repositories that make research artifacts available to the network or services that can add value to those artifacts. These nodes communicate according to a variety of patterns. A typical pattern is conversational, whereby the repository requests the provision of a service for a specific artifact, and the service relays the result of providing it. Such a service can be anything that adds value to an artifact, including, for example, registration in the scholarly record, computation of a trusted content-based hash, peer-reviewing, long-term archiving, and extracting citations from a textual artifact. In another pattern, the repository volunteers information pertaining to one of its artifacts in an effort to enhance the accuracy and up-to-datedness of information held by other nodes in the network. Such communications are unidirectional and include, for example, informing another artifact and its authors that they have been cited and announcing the relationship between a paper and a dataset to a knowledge graph in the cloud.

Interoperability Fabric: Event Notifications

Event Notifications: Overview

The interoperability fabric that enables a seamless interaction among these nodes in the network is based on Linked Data Notifications[3] (LDN) with payloads that adhere to a profile of the ActivityStreams2[4] vocabulary (AS2). It is formalized in the "Event Notifications in Value-Added Networks"[5] specification and described at the entry level in Hochstenbach et al. (2022). One can think of it as email intended for machine consumption. A notification that conveys a service request is sent from a repository to an LDN Inbox associated with a service. Behind the scenes, the service is performed, and eventually, the result is relayed to an LDN Inbox associated with the repository. Depending on the nature of the service, results can be delivered by value (inline in the notification) or by reference (as a link to a web resource). The notification payloads are intentionally lean, typically only containing URLs to identify the entities involved in the interaction. To yield further information about the artifact, the "follow your nose" auto-discovery principle is used. FAIR Signposting[6] allows discovering metadata and content

[1] https://www.theguardian.com/commentisfree/2019/mar/04/the-guardian-view-on-academic-publishing-disastrous-capitalism
[2] https://theconversation.com/increasing-open-access-publications-serves-publishers-commercial-interests-116328
[3] https://www.w3.org/TR/ldn/
[4] https://www.w3.org/TR/activitystreams-core/
[5] https://www.eventnotifications.net
[6] https://signposting.org/FAIR/

resources using the artifact's landing page[7] URL, WebID[8] supports the discovery of attributes pertaining to repositories, services, and actors, and the LDN protocol specifies auto-discovery of LDN Inboxes of entities involved. The asynchronous communication between nodes in the network specified by Event Notifications can be considered an atypical "API" because there never is a direct interaction with a repository or service API endpoint. Rather, the interactions are with a uniform mailbox associated with these nodes. The LDN mailbox is the API, allowing incoming notifications to be processed automatically and manually.

Table 1. A notification requesting the review of a scholarly artifact compliant with the COAR Notify and Event Notification specifications.

```
{
     "@context": [
       "https://www.w3.org/ns/activitystreams",
       "https://purl.org/coar/notify"
     ],
     "actor": {
       "id": "https://orcid.org/0000-0002-1825-0097",
       "name": "Josiah Carberry",
       "type": "Person"
     },
     "id": "urn:uuid:0370c0fb-bb78-4a9b-87f5-bed307a509dd",
     "object": {
       "id": "https://a.repository.org/preprint/921203/",
       "ietf:cite-as": "https://doi.org/10.73850/12345680",
       "type": "sorg:AboutPage",
       "ietf:item": {
         "id": "https://a.repository.org/preprint/921203/content.pdf",
         "mediaType": "application/pdf",
         "type": [
           "Link",
           "Article",
           "sorg:ScholarlyArticle"
         ]
       }
     },
     "origin": {
       "id": "https://a.repository.org/",
       "inbox": "https://a.repository.org/inbox/",
       "type": "Service"
     },
     "target": {
       "id": "https://review-service.com/system",
       "inbox": "https://review-service.com/inbox/",
       "type": "Service"
     },
     "type": [
       "Offer",
       "coar-notify:ReviewAction"
     ]
}
```

Implementations of this notification-based interoperability approach are currently in various stages of maturity. By far, the most advanced is the COAR Notify[9] effort, generously funded by Arcadia, which focuses on overlay review as a service that is made available for research papers hosted in repositories. It involves various significant parties representing repositories and review

[7] https://signposting.org/conventions/#scholobject. A landing page is a resource to which an artifact's persistent identifier resolves. In many cases, that is a web page presenting details about the artifact, including a description and links to metadata and downloadable content. In other cases, such as HTML journal articles, it represents the artifact's actual content.

[8] https://www.w3.org/2005/Incubator/webid/spec/

[9] https://www.coar-repositories.org/notify/

services, including bioRxiv[10], medRxiv[11], Dataverse[12], Zenodo[13], PCI Peer Community In[14], and Open Journal Systems[15]. The COAR Notify protocol[16] describes the possible communication patterns between repositories and services in great detail and fully aligns with the intentionally more generic Event Notification specification. In the EU DICE project, a prototype has been developed that allows requesting the ingestion of a dataset into a long-term archive[17]. In the Dutch collaboration "Netwerk Digitaal Erfgoed" (Network Digital Cultural Heritage), which deliberately tackles information interoperability challenges with a decentral mindset, the use of the notification-based approach has been explored for a variety of scenarios pertaining to dataset registration[18,19]. Experiments conducted in the context of the ResearcherPod project have explored repository-to-repository communication of dataset/paper links as an alternative to the centralized Scholix framework (Hochstenbach et al., 2022), citation extraction as a service, and citation notification as a service (Hochstenbach et al., 2023).

Table 1 shows an LDN/AS2 message that is compliant with the COAR Notify (and hence Event Notification) specification in which a review is requested for a fictitious artifact with the landing page `https://a.repository.org/preprint/921203/`, persistent identifier `https://doi.org/10.73850/12345680`, and PDF file at `https://a.repository.org/preprint/921203/content.pdf`. The identifier of the notification is `urn:uuid:0370c0fb-bb78-4a9b-87f5-bed307a509dd`.

Event Notification: Scenario

A decentralized scholarly communication network in which nodes that host a wide range of scholarly artifacts and nodes that provide a wide range of discrete and decoupled services (Priem & Hemminger, 2012; Van de Sompel et al., 2004) for those artifacts are interacting, empowered by the event notifications interoperability fabric, can best be explained by means of an example. The top and bottom of Figure 1 show nodes that host scholarly artifacts, respectively Repository A and Bob's Pod, the latter a personal data pod that Bob uses to share his scholarly work. Both hosting nodes are equipped with the capability to send and receive LDN notifications with AS2 payloads, which are compliant with the Event Notifications specification. The middle of Figure 1 shows three nodes that provide value-added services for scholarly artifacts that are made available to the network. At the left is a registration service, which makes a web resource officially part of the scholarly record and turns it into a scholarly artifact. This could, for example, be achieved by providing a persistent identifier, content-based hash, and/or trusted datetime for the resource. In the middle is an archiving service, which creates an archived version of a scholarly artifact in a long-term archive. And to the right is an awareness service. This could, for example, be a scholarly search engine or knowledge graph, but it could also be a service that merely relays incoming information to such services. In an Event Notification network, all services have the capability to send and receive LDN/AS2 notifications, implementing the asynchronous communication patterns as specified by the protocol.

The scenario starts with Alice depositing a scholarly contribution to Repository A, where it becomes available on the web via a landing page at URL-A. Following the deposit, services are asked to add value to Alice's contribution, one by one. Depending on the nature of the

[10] https://www.biorxiv.org
[11] https://www.medrxiv.org
[12] https://dataverse.org
[13] https://zenodo.org
[14] https://peercommunityin.org
[15] https://openjournalsystems.com
[16] https://notify.coar-repositories.org
[17] https://dans-labs.github.io/ddps-docs/
[18] https://erfgoedpod.github.io/usecases/
[19] https://netwerkdigitaalerfgoed.nl/wp-content/uploads/2023/07/20230614-Miel-Vander-Sande_PLDN-Solid-Use-Cases.pdf

repository and the nature of the service, a request for service can be triggered automatically or manually:

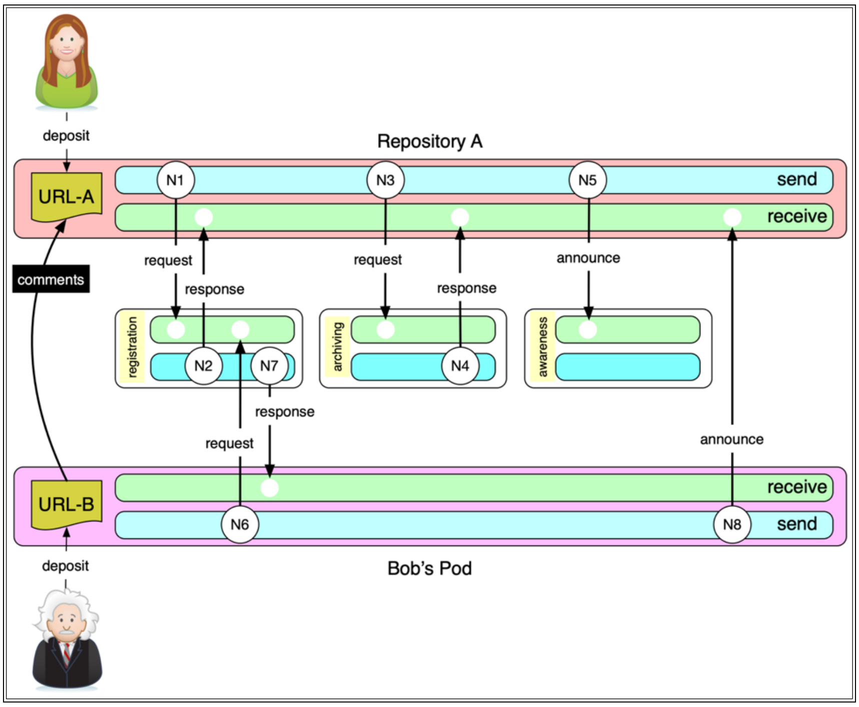

Figure 1. A scenario illustrating a scholarly communication network empowered by Event Notifications.

1. N1 – Notification N1 is sent to a registration service. It critically contains URL-A, Alice's unique web identifier, as well as an indication of the requested service (i.e., registration);

2. The registration service receives notification N1 and uses Signposting and WebID auto-discovery techniques to obtain further information (e.g., metadata) about Alice's contribution at URL-A and about Alice herself. Depending on how the service implements registration, it may also obtain the contribution itself, for example, to create a content-based hash for it;

3. N2 – The registration service starts some internal registration workflows (e.g., a human cataloguing task) and, after some time, responds to the registration that was requested with notification N1 by sending notification N2. Since a registration service is concerned, most likely, the result (e.g., a persistent identifier) is delivered by value, inline in the notification;

4. Repository A receives notification N2 and can add the result of the registration service to the record for Alice's contribution, which has now officially become a scholarly artifact;

5. N3 – To save Alice's artifact for posterity, a request to archive it is sent in notification N3, which, again, contains pertinent information as mentioned above regarding N1;

6. The archiving service receives notification N3 and adds Alice's artifact at URL-A to the queue of resources that need to be archived. Eventually, using web-archiving techniques and helped by Signposting, all relevant resources pertaining to Alice's artifact are pulled into the archive;

7. N4 – The archiving service sends notification N4 to Repository A, indicating that Alice's artifact has been archived. The service result is provided by reference as the URL of the Memento that the archive created for URL-A. Via that Memento, the Mementos of all other resources pertaining to Alice's artifact can be reached;

8. Repository A receives notification N4, and flags Alice's artifact as having been archived, for good measure adding the URL of the Memento of URL-A to its record;

9. N5 – Now that Alice's contribution has been registered and archived, it is time to broadcast its existence so other researchers can peruse it. To that end, Repository A sends notification N5 to announce the new artifact to the awareness service. Again, the notification contains the pertinent information as mentioned before. Since an announcement is considered, Repository A knows not to expect a response;

10. The awareness service receives notification N5. Since it is a full-text search engine, it uses URL-A and the Signposting available there to obtain all pertinent resources and index them.

Bob discovers Alice's artifact while searching the awareness service. He likes what he reads and decides to volunteer some additional ideas for future work. He launches his favorite editor, writes down his ideas, and posts them in his pod as a document at URL-B. Since Bob would like to get scholarly credit for these ideas, he officially registers them and then informs Alice about his feedback:

11. N6 – Notification N6 is sent to the registration service. It critically contains URL-B and Bob's unique web identifier;

12. The registration service receives notification N6 and proceeds in the same way as it did for the registration request for Alice's contribution;

13. N7 – The registration service responds to the registration that was requested with notification N6 by sending notification N7;

14. Bob receives notification N7 and adds registration information to the record of his contribution;

15. N8 – Now that his feedback is officially registered, Bob announces it in notification N8 to Repository A;

16. Repository A relays the information to Alice and adds a link to Bob's comments to the record for Alice's artifact.

Event Notification: Consequences

To a large extent, when a scholarly artifact goes through a value chain, the current system for scholarly communication does not record the relationship between the artifacts prior to and post such a value chain[20]. Because such relationships, for example, between a preprint and its peer-reviewed version, are not recorded at the moment, they must be reverse-engineered, with great pain, at some later stage (Besançon et al., 2023; Cabanac et al., 2021). Such reverse engineering, which has been the go-to approach for over two decades, is compute-intensive, heuristics-based, maintenance-intensive, error-prone, typically centralized, and results in linkage information becoming available with significant delays.

A major consequence, and actually motivator, for the conversational nature of the repository/service communication enabled by Event Notifications is the ability to interlink an

[20] https://doi.org/10.5281/zenodo.8076843

artifact and a service result in near real-time, bi-directionally. This aspect of Event Notifications addresses the aforementioned long-standing problem. Indeed, the receiver of a service request knows which artifact the request is for; hence, its service result – if provided by reference as a web resource – can link to it. The receiver of such a service result can link to it from the artifact for which the service was requested.

Another consequence is the experimental confirmation of the premise of COAR's Next Generation Repositories report that the addition of interoperability affordances can turn repositories into active nodes in the scholarly communication network. Actually, Event Notifications support the emergence of a thriving, network-wide point-to-point dialogue among repositories and services. In the aforementioned Scholix-related experiment, dataset/paper linkage information involving artifacts hosted by Belgian institutional repositories was communicated by means of LDN/AS2 messages to synthetic repositories that hosted the artifacts at the other end of each linkage. For three Belgian repositories, this involved communicating with 635 other repositories (Hochstenbach et al., 2022). In the aforementioned citation/extraction experiment, 1896 citations were extracted from 100 papers from a single institutional repository, resulting in sending notifications to 163 (synthetic) repositories (Hochstenbach et al., 2023).

Event Logs: Transparency Regarding an Artifact's Life Cycle

The previous sections provided insights into how the Event Notification fabric can provide transparency in the scholarly communication process by exchanging information about value-adding services. This section highlights the potential for further transparency by publishing Event Notifications as a persistent resource. Event Notifications published as append-only Event Logs provide the potential for all actors in the network to discover the provenance information of all value-adding events pertaining to an artifact. Given an artifact URL, an actor can find the location of the Event Logs using auto-discovery mechanisms and inspect the life cycle. Figure 2 illustrates Event Logs that result from the scenario depicted in Figure 1:

- The Event Log for Repository A contains the notifications pertaining to Alice's artifact sent to and received from the registration (N1, N2), archiving (N3, N4), and awareness (N5) services as well as the notification (N8) received regarding Bob's comment;
- The Event Log for the registration service contains the notifications it received regarding Alice's (N1) and Bob's (N6) contributions as well as the respective responses (N2 and N7);
- Similarly, the Event Logs of the other services contain entries regarding Alice's artifact (N3 and N4 for the archiving service; N5 for the awareness service);
- The Event Log for Bob's comment contains entries for its registration (N6, N7) and an entry for the announcement sent to Repository A.

Data nodes in the network can provide Event Logs that detail all events pertaining to their artifacts, allowing client applications to obtain immediate insight into the life cycle events that each artifact underwent. In the same way, service nodes can provide Event Logs for the services they provide for artifacts. As such, Event Logs can provide transparency regarding the life cycle of scholarly artifacts. To fulfill this function, Event Logs need to support discovery; this aspect is discussed in the section "Event Logs: Discovery". Also, if assessments regarding artifacts are going to be based on the life cycle events recorded in their respective Event Logs, the trustworthiness of these logs is crucial; this aspect is discussed in the section "Event Logs: Trust".

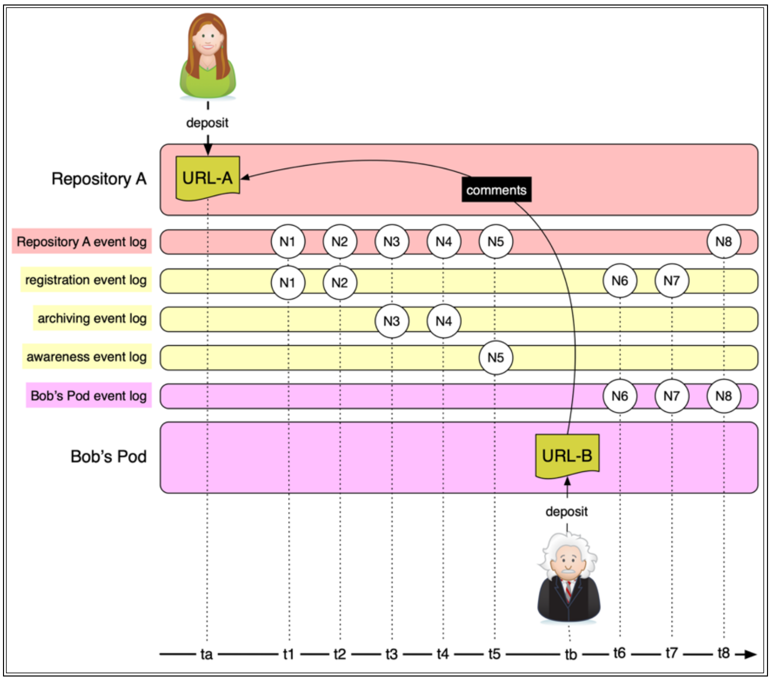

Figure 2. Event Logs resulting from the scenario depicted in Figure 1.

Event Logs: Discovery

Since data and service nodes come in different shapes and sizes, flexibility regarding the technical implementation of Event Logs may be necessary. For example, a personal data pod that hosts a few hundred artifacts might find a file-based implementation (e.g., an Event Log per artifact) appealing. In contrast, a popular service node might prefer to store all events in a database. Yet, irrespective of the Event Log implementation, the discovery of event information should be uniform and, at least, the discovery of all events for a given artifact should be supported by both data and service nodes. This could be achieved by using the following Web Linking (RFC8288) approach that uses HTTP headers to make the location of Event Logs discoverable:

- Define a link relation type (e.g., `eventlog`) for which the link target is an Event Log;

- In responses to HTTP HEAD/GET requests to URLs at the end of data nodes (e.g., artifact URLs, LDN Inbox URL) and service nodes (e.g., LDN Inbox URL), provide a Link-Template[21] HTTP header field that expresses the link template for the discovery of the events for a given artifact. As an example, the following template could be used: `Link-Template: "/events/{artifact}"; rel="eventlog".`

Event Logs: Trust

In the current scholarly communication system, the sense of trust is provided by the mere reputation of a node in the network. A trustworthy node ensures authentic contributions and added-value to the scholarly record and keeps those accessible for the long-term in a tamperproof way. Reputational damage, for instance, in cases when fraudulent behavior is detected (see, for example, Abalkina, 2021; van Noorden, 2023), can diminish trust or even result in its loss. Currently, extensive data analysis work by scientific sleuths is required to discover such reliability breaches. Additionally, anticipating a scholarly communication system

[21] https://datatracker.ietf.org/doc/draft-ietf-httpapi-link-template/

in which researchers make their contributions available in personal data pods, one cannot expect trust to merely derive from reputation. As such, trust that can objectively be verified based on interactions in the system becomes a requirement.

These insights motivated our work into adding machine verifiable trust to Events Logs, a first step in making trust in the value chain explicit and making abuses easier to trace. Table 1 provides an overview of various aspects of a notification-driven communication system for which adding verifiable trust could be considered.

Table 2. Aspects in an Event Notification network for which verifiable trust could be added.

Trust levels	**Challenges**
Trust regarding the notification	C1.1 Proof regarding the veracity of the notification (e.g., is the received notification identical to the sent notification)
Trust regarding the sender	C2.1 Proof that the sender created the notification. C2.2 Proof the sender's trustworthiness (e.g., filter out bad actors)
Trust regarding the Event Log	C3.1 Proof of the authenticity of events in the Event Log (e.g., notifications in the Event Logs are as they were received). C3.2 Proof of the local completeness of the Event Log (e.g., no notifications were removed or manufactured). C3.3 Proof of the global completeness of the Event Log (e.g., the Event Log contains all globally known notifications about the artifact)
Long-term Event Log trust	C4.1 Providing long-term proofs of Event Logs (e.g., using archived artifacts and Event Logs)

Trust regarding the notifications (C.1.1) requires some hashing technology to prove that the received notification is identical to the sent notification. Technologies such as HTTP Digest headers[22] can be explored to include the hash of the LDN/AS2 payload in every notification. These hashes range from straightforward SHA checksums of the serialized JSON-LD payloads to more complex methods such as RDF Dataset Canonicalization[23] that are not dependent on the serialization method.

Trust in the sender (C.2.1) may be attained by using one of these methods:

- Digital signatures: With a digital signature, a sender can prove to be in the possession of the private key that produced a valid signature of an LDN/AS2 message. HTTP Signature headers[24] or Linked Data Signatures[25] can be used to transport these signatures in the network;
- Authentication layer: By adding an authentication layer such as OIDC[26] (e.g., with ORCID[27] or WebIDs[28]) on top of the LDN protocol, it is possible only to consider incoming notifications from well-known senders, i.e., senders with verifiable identities and/or attributes (C.2.2). Group WebIDs, as used in Solid-OIDC[29], could provide a

[22] https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-digest-headers-13
[23] https://www.w3.org/TR/rdf-canon/
[24] https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-message-signatures-19
[25] https://w3c.github.io/vc-data-integrity/
[26] https://openid.net/specs/openid-connect-core-1_0.html
[27] https://orcid.org
[28] https://www.w3.org/2005/Incubator/webid/spec/
[29] https://solidproject.org/TR/oidc

low-tech federation alternative to current SAML-based federated identity management systems.

Trust regarding the Event Log requires some form of guarantee that events were not fabricated or modified by some actor (C3.1). Digital signatures can prove that each entry in the Event Log is authentic, but the verification process necessitates the reliance on long-term secrets (e.g., private keys) in the network. The loss of a secret makes verification of notifications signed with that secret unverifiable, undermining the intended trust. An extensive set of technologies can be imagined that mitigate against this loss of secrets or at least minimize the effects of loss.[30] But, to quote Diffie (2003): “The secret to strong security: less reliance on secrets”. For Event Logs, a more pragmatic approach would be to explore technologies that don’t rely on the long-term availability of secrets but rather on the redundancy of Event Log entries in the network. At least the service node and data node should both keep a public copy of LDN/AS2 messages in their respective Event Logs. Web-archiving solutions, such as LOCKSS[31], can then distribute Event Logs over many archival nodes so that large-scale manipulation becomes expensive. A data or service node could passively make archiving of Event Logs possible by publishing a Sitemap that – per artifact – lists the URL where all its events can be retrieved. Or nodes could proactively archive by submitting those URLs to web archives (via the archive’s API or via an event notification that requests archiving) as soon as a new event for an artifact occurs. As a matter of fact, the temporal delta between the occurrence of an event for a given artifact and the archiving of the list of all events for that artifact might serve as an extra indicator of trust in a node: the faster archiving occurs, the smaller the chances that tampering occurred.

As a side note, these considerations were also informed by the choices that ActivityPub[32] made for their outbox mechanism, which is a log of all outgoing notifications and, as such, a subset of the messages that an Event Log would contain. ActivityPub requires HTTP URIs for each notification entry in the outbox, allowing their retrieval by dereferencing the URI. Applying this approach to Event Notifications would require a non-trivial commitment by data nodes and service nodes to keep notification URIs operational long-term. By redundantly storing Event Logs such a long-term commitment is delegated to third parties, such as web archives, that have an inherent commitment to longevity.

While trust in individual entries in an Event Log could be established as discussed above, a challenge remains to ensure that the Event Log in its entirety is tamperproof, i.e., that no legitimate entries were removed or manufactured ones were added (C3.2). The existence of the same entries in Event Logs of data nodes and service nodes that have interacted regarding an artifact, combined with the redundant storage of Event Logs achieved via web archives, could address C3.2 and C4.1. Event Logs could be serialized using some fragmentation techniques, such as Linked Data Event Streams[33]. For each stable fragment checksums can be calculated (or compared) at the side of the web archive, providing trust by a third party.

Event Logs may also fail to capture all value-adding events when data nodes are not “kept in the loop” about all value-adding actions pertaining to their artifacts (C3.3). This situation can arise, for instance, when a data node requests service node S1 to provide a service for one of its artifacts and S1 delegates the service provision to service node S2. When, as part of the ensuing communication between S1 and S2, the data node is not updated about S2’s service result, the artifact’s Event Log on the data node will be incomplete. To prevent this from happening, some extra instructions can be added to the specifications that clearly state who should receive an event notification. For instance, in all communication about any artifact, a copy of the message should be sent to the LDN Inbox of the artifact (which can be discovered by resolving the

[30] See, for example, the security and privacy considerations in https://w3c.github.io/vc-data-integrity/

[31] https://www.lockss.org

[32] https://www.w3.org/TR/activitypub/

[33] https://semiceu.github.io/LinkedDataEventStreams/

artifact URL and checking its HTTP headers). An alternative solution could be in the form of aggregator services that crawl the web in search of this out-of-band information and update data nodes accordingly. Global completeness of an Event Log would still require navigating the global web, affordances can be used to ease the discovery and processing of added-value events.

The focus in this section was on the trustworthiness of Event Logs and did not touch on trust regarding the artifacts that are the subject of the notifications. It should be noted that, in the established scholarly system, this type of trust is also derived from the reputation of a network node. For instance, a trusted publisher is expected to provide a range of guarantees in an integrated manner, including accurately timestamping research artifacts, organizing credible peer-review, ensuring the fixity of artifacts or providing a provenance trail when they change, and ensuring long-term access (for challenges with this aspect see, e.g., Van Noorden, 2023; Cabanac, 2024; Eve, 2024). In a decentralized system, such aspects of trust in an artifact are provided in a distributed rather than integrated manner, i.e., different parties provide different guarantees. Trust providers could include timestamping services as per RFC3161[34], providers of content-based checksums as used in distributed version control systems such as Git[35], overlay peer-review/certification nodes, and distributed/interoperable web-archiving solutions.

Conclusions

Our envisioned, alternative, decentralized scholarly communication value chain leverages widely accepted web standards, allowing for cost-effective implementation and maintenance using existing infrastructure. Existing systems that make scholarly artifacts available on the web, such as publishers, institutional and discipline repositories, data repositories, and software platforms, can act as data nodes. However, novel systems such as personal researcher pods can also be used. Value is added to artifacts by distributed nodes that provide specific services, including those that fulfill the core functions of scholarly communication, i.e., registration, certification, awareness, and archiving. Value can also be added by other actors in the network, such as researchers who annotate, comment, or reference a colleague's artifact.

The glue that loosely connects the nodes that make artifacts available with those that add value to them is the point-to-point Event Notifications protocol that, intentionally, facilitates rapid interlinking of an artifact with its value-added resources. The protocol specification is stable, has been thoroughly prototyped, and is currently being implemented by significant parties in the COAR Notify and "Netwerk Digitaal Erfgoed" efforts. The former focuses on an environment in which data nodes are institutional/discipline repositories and services nodes provide overlay peer-review. It constitutes an important and concrete step on the path toward realizing COAR's Next Generation Repository vision to create an alternative decentralized scholarly communication system in which repositories play a proactive role in the scholarly communication process.

The Event Logs work anticipates a scholarly communication network in which Event Notifications continuously flow among data and service nodes. They are conceived to provide full transparency regarding the life cycle of artifacts. To fulfill that role in a credible manner, they must be accurate, complete, and trustworthy. A variety of activities aimed at determining which technologies can be used to that end are underway. For each of the challenges that were discussed above, finding an approach with a low implementation barrier is a guiding principle.

Overall, the Event Notifications and Event Logs work aligns with other ongoing efforts that aim at bringing about a different research communication system that truly leverages the affordances of the network environment, is beneficial to the research community and society at large, and is

[34] https://datatracker.ietf.org/doc/html/rfc3161
[35] https://en.wikipedia.org/wiki/Git

not at the mercy of the monopoly of the large publishers. As Nottingham (2023) states, centralization must not necessarily lead to harmful consequences, but quite commonly, it does. And indeed, while the reputation of established publishers may provide a guarantee that the artifacts they publish can be trusted, it has become increasingly clear that they can't be trusted when it comes to the data they collect while surveilling users (Posada & Chen, 2018; Gatti, 2020; Siems, 2021; Pooley, 2022; Yoose & Shockey, 2023). As such, it is better to avoid further harm caused by centralization. Working on building blocks for a decentralized research communication network is one way to do so.

Acknowledgments

This work is funded by the Andrew W. Mellon Foundation (grant number: 1903-06675) and supported by SolidLab Vlaanderen (Flemish Government, EWI, and RRF project VV023/10). The authors would like to thank the ErfgoedPod project (Miel Vander Sande) and IMEC (Ruben Dedecker) for joint discussions on the requirements for Event Notifications and Event Logs in both of our projects. The authors would also like to thank the COAR Notify (Kathleen Shearer, Paul Walk, and Martin Klein) for involving them from the outset, allowing them to provide input and gain valuable insights in the scholarly communication process.

References

Abalkina, A. (2021). Detecting a network of hijacked journals by its archive. *Scientometrics*, *126*, 7123–7148. doi.org/10.1007/s11192-021-04056-0

Besançon, L., Cabanac, G., Labbé, C., & Magazinov, A. (2023). Sneaked references: Cooked reference metadata inflate citation counts. *arXiv*. doi.org/10.48550/arXiv.2310.02192

Bollini, A., Knoth, P., Perakakis, P., Rodrigues, E., Shearer, K., Van de Sompel, H., & Walk, P. (2017). Next generation repositories report. Confederation of Open Access Repositories. doi.org/10.5281/zenodo.8077381

Cabanac, G. (2024). *Fake science : panorama des méconduites et contre-feux pour déjouer les pièges.* Retrieved from https://ut3-toulouseinp.hal.science/hal-04225515

Cabanac, G., Oikonomidi, T., & Boutron, I. (2021). Day-to-day discovery of preprint–publication links. *Scientometrics*. doi.org/10.1007/s11192-021-03900-7

Diffie, W. (2003, January). Perspective: Decrypting the secret to strong security. *News.com*. Retrieved from http://web.archive.org/web/20040929082510/http://news.com.com/2010-1071-980462.html

Eve, M. P. (2024). Digital scholarly journals are poorly preserved: A study of 7 million articles. *Journal of Librarianship and Scholarly Communication, 12*(1). doi.org/10.31274/jlsc.16288

Gatti, R. (2020). *Business models and market structure within the scholarly communications sector.* International Science Council. doi.org/10.24948/2020.04

Larivière, V., Haustein, S., & Mongeon, P. (2015). The oligopoly of academic publishers in the digital era. *PLOS ONE*, *10*(6). doi.org/10.1371/journal.pone.0127502

Hochstenbach, P., Verborgh, R., & Van de Sompel, H. (2023). Koreografeye: An event-driven orchestrator for scholarly value chains. *Code4Lib Journal*, *58*. Retrieved from https://journal.code4lib.org/articles/17823

Hochstenbach, P., Van de Sompel, H., Vander Sande, M., Dedecker, R., & Verborgh, R. (2022). Event notifications in value-adding networks. In G. Silvello, O. Corcho, P. Manghi, G. M. Di Nunzio, K. Golub, N. Ferro, & A. Poggi (Eds.), *Linking Theory and Practice of Digital Libraries* (pp. 133–146). Springer International Publishing. doi.org/10.1007/978-3-031-16802-4_11

Nottingham, M. (2023). *Centralization, decentralization, and internet standards* (RFC 9518). Retrieved from https://www.rfc-editor.org/rfc/rfc9518.html

Pooley, J. (2022). Surveillance publishing. *The Journal of Electronic Publishing*, *25*(1). doi.org/10.3998/jep.1874

Posada, A., & Chen, G. (2018). Inequality in knowledge production: The integration of academic infrastructure by big publishers. *ELPUB 2018,* Jun 2018, Toronto, Canada. doi.org/10.4000/proceedings.elpub.2018.30

Priem, J., & Hemminger, B. (2012). Decoupling the scholarly journal. *Frontiers in Computational Neuroscience*, *6*(19). doi.org/10.3389/fncom.2012.00019

Siems, R. (2021, April). When your journal reads you. *Elephant in the Lab*. doi.org/10.5281/zenodo.4683778

Van de Sompel, H., Payette, S., Erickson, J., Lagoze, C., & Warner, S. (2004). Rethinking scholarly communication: Building the system that scholars deserve. *D-Lib Magazine*, *10*(9). doi.org/10.1045/september2004-vandesompel

van Noorden, R. (2023, December 12). More than 10,000 research papers were retracted in 2023 - a new record. *Nature*. doi.org/10.1038/d41586-023-03974-8

Yoose, B., & Shockey, N. (2023). Navigating risk in vendor data privacy practices: An analysis of Elsevier's ScienceDirect. SPARC. doi.org/10.5281/zenodo.10078610