TRANSPARENT AND TRUSTWORTHY ARTIFACT LIFE CYCLE DATA

Patrick Hochstenbach
Ghent University Library
Belgium
patrick.hochstenbach@ugent.be
0000-0001-8390-6171

Martin Klein
Los Alamos National Laboratory
USA
mklein@lanl.gov
0000-0003-0130-2097

Herbert Van de Sompel
DANS
The Netherlands
hvdsomp@gmail.com
0000-0002-0715-6126

Ruben Verborgh
Ghent University - IMEC
Belgium
ruben.verborgh@ugent.be
0000-0002-8596-222X

Abstract **– In scholarly communication, alternative publication platforms are rising, offering services that are not tied to and function independently from traditional publishers. Decentralized and decoupled versions of value-adding services such as overlay journals, peer review and endorsement services are provided to the network. These alternatives address not only the massive monopolization of scholarly communication but also new demands for rapid distribution of scholarly artifacts in fields such as life sciences. They also accommodate for new publication formats such as data sets and source code that are beyond the traditional paper. Event Notifications provide an asynchronous point-to-point messaging protocol that acts as an interoperability layer between the nodes in such networks. Event Logs are a proposal to provide full transparency in the scholarly process by publishing public resources on each node containing lifecycle information pertaining to the scholarly artifacts in the network. Trust in Event Logs require some form of machine-verifiability because trust by reputation of each node is not scalable in large networks. This paper presents a pragmatic approach to building trust without relying on costly blockchain technologies, which lack sustainable long-term strategies. By using trusted archival nodes within Event Notification networks and Event Logs, we propose a decentralized, transparent, and trustworthy alternative to traditional scholarly communication systems.**

Keywords **– Scholarly communication, decentralized web, digital repositories, digital preservation, network protocols, trust, transparency.**

I. INTRODUCTION

The web-based scholarly communication landscape has seen noticeable changes in the recent past. For example, the COVID-19 pandemic has accelerated the adoption of preprints in domains such as the life sciences, where this rapid public distribution of manuscripts before undergoing peer review was not particularly common in the past [1,2]. In addition, scholarly artifacts, beyond the traditional paper in PDF format, such as datasets and source code are being published, in dedicated venues, and are slowly being considered in the context of institutions' and scholars' performance evaluations [3-6]. A third example of the evolution in the scholarly communication domain is the emergence of decentralized value-added services such as overlay journals, stand-alone peer review and endorsement services, and other entities that embrace the publish-review-curate (PRC) model, decoupled from traditional publishing services [7].

The community also has, for years now, observed the trend of significantly increasing numbers of publications [8]. The powers of artificial intelligence (AI), specifically generative AI models, which we are

iPRES 2024: The 20th International Conference on Digital Preservation, Ghent & Flanders (Belgium).

DOI: 10.1145/nnnnnnn.nnnnnnn

merely beginning to comprehend, are further highlighting the reality that producing somewhat reasonable scholarly articles en masse is far from impossible.

While most of these evolutionary aspects are generally considered rather positive, they also, in part, emphasize the ongoing crisis in scholarly communication. Specifically, we learn that traditional peer review models are unsuited for these new quantities and types of scholarly artifacts [9,10]. The resulting lack of quality validation leads in part to a replicability and reproducibility crisis and without the ability to validate, we lose trust in our scholarship.

The remainder of this paper is structured as follows: In Section II we examine the current state of art to provide an alternative decentralized scholarly communication infrastructure, alongside identifying the prerequisites for exposing transparent and trustworthy artifact life cycle data. In Section III we introduce how transparency and trust regarding that life cycle data can be achieved by means of a pragmatic approach based on redundant web archiving. In Sections IV and V we sketch the interactions between an institutional repository that hosts a (scholarly) artifact and a peer review system that creates a peer review for this artifact and how these systems generate event logs containing life cycle data. We document the payloads of the asynchronous messaging system required in our network in Section VI. Section VII describes how web agents can verify the trustworthiness of artifact life cycle data. In Section VIII we offer a discussion of our results and possible routes for implementation. We offer conclusions and some aspects of future work in Section IX.

II. Background

In previous work, we argued in favor of establishing novel levels of transparency regarding the scientific process that aim to increase trust and support reproducibility [11]. Technically, the approach enables an alternative, decentralized scholarly communication network that consists of interacting *data nodes* that host (research) artifacts and *service nodes* that add value to these artifacts. The "Event Notification in Value-Adding Networks" specification [12] provides interoperability affordances for the nodes in the network to communicate in an asynchronous and point-to-point way about life-cycle information pertaining to artifacts. The protocol inherently adopts an asynchronous design, necessitated by the unpredictable duration required for the provisioning of value-added services by service nodes. The time frame for providing such services can range anywhere from a few seconds (for instance, generating a trusted timestamp) to several months (notably in scenarios requiring a peer review). To accommodate this variability the protocol introduces Linked Data Notification (LDN) Inboxes, which are similar to mailboxes, as the communication point for data nodes and service nodes. In this communication channel, Event Notifications with ActivityStreams2 (AS2) payloads, expressed as JSON-LD, are exchanged. This push-based protocol obviates continuous pulling for information about a value-added service result.

The Event Notification in Value-Adding Network protocol allows communities to create profiles for their specific use-cases. By far the most advanced implementer is the COAR Notify effort [7], generously funded by Arcadia, that focuses on providing peer-review services for a network of repositories. Pilot projects started in 2022 to implement the COAR Notify protocol and currently involve significant partners such as bioRxiv[1], medRxiv[2], Dataverse[3], Zenodo[4], PCI Peer Community In[5] and Open Journal Systems[6]. Quite recently, in July 2023, DSpace started to include support for the COAR Notify protocol in version 8 of the repository software [13].

In "Event Notifications and Event Logs: Transparent Sharing of Artifact Life Cycle Data" [14], we argue for additional transparency in the scholarly communication process by publishing Event Logs for each artifact that list all event notifications that were exchanged about it. These Event Logs could provide full transparency about the value-added service that were provided for an artifact including life cycle information about how an artifact is registered, reviewed, published, indexed, archived, etc.

[1] https://www.biorxiv.org
[2] https://www.medrxiv.org
[3] https://dataverse.org
[4] https://zenodo.org
[5] https://peercommunityin.org
[6] https://openjournalsystems.com

In the current scholarly communication system, the metadata for these value-added events are hard to find and can only be revealed by means of post-factum heuristic-bound mining processes across many research corpora [15,16]. Event Logs are envisioned as decentralized web resources that provide near-real time information about the life cycle of contributions to the scholarly record.

Also, in the current system, the core functions of scholarly communication (registration, certification, awareness, and archiving) are very much centralized inside publisher nodes. Trust in these nodes is provided by reputation. Trustworthy nodes should ensure only authentic value-added contributions are added to the scholarly record and preserved for the long-term in a tamper-proof way. Reputation can be gained, by a good track record of services, but also lost when fraudulent behavior is detected [17]. In a decentralized network, containing thousands of repository nodes and potentially millions of researcher homepages, trust can't be expected to be derived from reputation alone and some form of machine-assisted verifiability is required. Although Event Logs are still a work in progress, and no specification is available yet, our aforementioned paper introduced some new aspects of verifiable trust that should be provided in decentralized networks. Three primary examples of trust involve: confidence in the Event Logs to accurately reflect veritable value-added events; the assurance that Event Logs are *authentic*, i.e. no fake value-added events were added; the assurance that Event Logs are *local* complete, i.e. append only, with no events being removed or edited; and an assurance that Event Logs are *global* complete, i.e. with no censorship of value-adding events by data and service nodes.

III. Proposed Approach

Digital signatures could provide a mechanism to prove that value-added events added to Event Logs are authentic, but verification of digital signatures requires the reliance on long-term secrets. The loss of secrets, or of the underlying infrastructure, prohibits long-term verification of the veracity of Event Logs. Block-chain technologies may mitigate some of these risks but come with high implementation costs [18] (not only monetary) or have thus far not demonstrated reliable long-term strategies. We argue for a pragmatic approach and explore solutions that do not rely on long-term secrets but rather generate redundancy of Event Logs in the network as a means to provide trust.

Since Event Logs are regular web resources, in this paper, we focus on using publicly and openly available archiving infrastructure, such as web archives or resource versioning systems that operate under an archival regime, to distribute Event Logs to create redundant read-only copies. Given a public Event Log on a decentralized node, it should be possible to verify its authenticity by consulting the archived versions, and the authenticity of the entries by comparing Event Logs across data nodes and service nodes.

The nodes that can keep read-only copies of Event Logs for the long-term are called *archive services* in this paper and are required: a) to be trustworthy, b) keep records in an immutable state, and c) implement the Event Notification and Memento protocol (RFC 7089)[19]. We believe that utilizing multiple such archives increases our chances for long-term availability of and access to these resources. We introduce a framework for pro-actively archiving Event Logs and interlinking artifacts with their respective Event Logs and with the archived copies thereof (mementos) accessible via Memento *TimeMaps*. This effort results in an environment where a research artifact is linked to Event Logs describing all relevant changes the artifact has undergone, and the corresponding archival records. We believe that this setup significantly increases transparency - and therefore trust - and is backed by a decentralized archiving infrastructure, which comes with an increased guarantee of persistence.

The next section introduces the actions of a peer review service that produces a review for a scholarly artifact URL-A that is hosted by a repository. To provide transparency in the life cycle of the scholarly artifact, information about the value-adding peer review service is published in a public Event log URL-SN-E. Trust is provided for this Event Log by creating an authentic memento of this log in a trusted archive.

In section V, a similar scenario is presented, now from the side of the repository that receives information about the new peer review. The repository also creates a public Event Log, URL-DN-E, containing life cycle information about the value-

adding peer review for scholarly artifact URL-A. Also here, trust is provided for this Event Log by creating an authentic memento in trusted archives. But, in this scenario the repository chooses to use an external *Choreographer* service as a relay to contact many archives (a local implementation choice).

All hosts in the network use the asynchronous messaging services provided by the Event Notifications protocol to update each other about value-adding events (such as peer review and archiving). An Event Notification 'Offer' notification is used to request a value-added service. An Event Notification 'Announce' notification is used to receive information about a value-added service result. Section VI provides the details of the actual Event Notification messages that are used in the network.

IV. Peer Review Service Archiving its Event Log

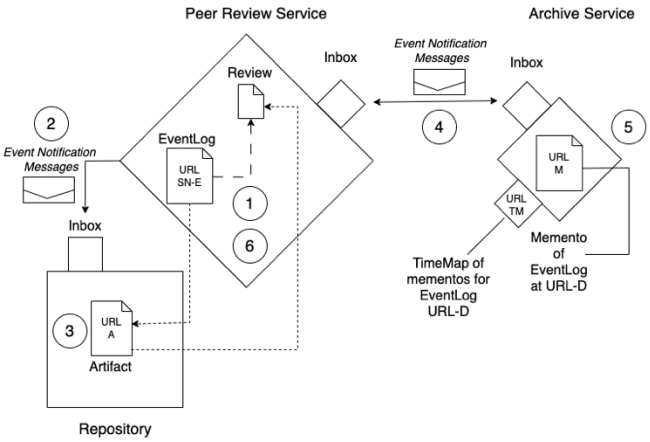

Figure 1. A peer review service sends an Event Notification message to a repository to announce the location of a new review for an artifact hosted on the repository. In parallel, the peer review service updates its Event Log and requests and receives a link to the synchronized memento for the updated Event Log from an archive service.

In this scenario, a peer review service creates a review about a (scholarly) artifact at URL-A that is hosted by a repository. The peer review service maintains a public Event Log at URL-SN-E that includes the value-added service that was provided for URL-A. To make this Event Log more trustworthy, the peer review service uses the services of an archive service to create a synchronized memento (URL-M) of its Event Log. Figure 1 sketches a six-step process to inform the network about the availability of a new peer review. In the remainder of this section, we will provide a walk-through of this process.

Step 1. The peer review service (the tilted square at the top of Fig. 1) creates a new service result: a review for artifact URL-A hosted at the repository (the square at the bottom left of Fig. 1). The peer review service updates the public Event Log URL-SN-E to include the value-added service provided for artifact URL-A. The new Event Log entry expresses the fact that a peer review was created for the artifact URL-A.

Step 2. The peer review service sends an Announce Event Notification message to the repository. This message informs the repository that a peer review is available for the artifact URL-A.

Step 3. The repository uses the information in the Announce message for some internal bookkeeping (e.g., adding a link to the peer review in the landing page of the artifact) and to update the local Event Log of the artifact URL-A with a new entry pertaining to the peer review. See Section V, for more information about this process.

Step 4. In parallel to step 2, the peer review service sends an Offer Event Notification to one or more archive services (the tilted square on the right for Fig. 1). The Offer notification requests archiving of the Event Log URL-SN-E pertaining to the review.

Step 5. The archive service uses the information in the Offer Event Notification message to create a memento URL-M for the Event Log URL-SN-E. The creation of the memento could take some time. When the memento is available, the archive service sends an Announce Event Notification message back to the peer review service, containing information about the location of the TimeMap for the Event Log URL-SN-E.

Step 6. The peer review service uses the information contained in the Announce Event Notification message to add a Link HTTP header to the Event Log URL-SN-E with the *TimeMap* location of the archive service (if no such link is already available). A TimeMap is defined by RFC7089 as the resource that lists the URIs or resources (mementos) that encapsulated prior states of the original resource (Event Logs in our case). By adding a TimeMap Link HTTP header to the Event Log, we provide web agents an affordance to access and verify versioned read-only copies of the Event Log at an archive service.

An example of such a TimeMap Link HTTP header is provided below:

```
Link: <http://archive.node/map/URL-SN-E>;
rel="timemap"
```

As a result of these six steps, the following resources are created or updated, concerning the Event Log of the peer review service:

- A new Event Log entry at URL-SN-E, containing information about a peer review provided for the artifact at URL-A.
- An archived copy of URL-SN-E, the memento of the Event Log entry, at URL-M, available in the TimeMap URL-TM.
- A Link HTTP header added to the URL-SN-E providing the location of the memento TimeMap of the archive service.

V. Repository Using a Choreographer to Archive the Event Log

In this scenario, the repository hosts a scholarly artifact URL-A. We assume that the repository will outsource the creation of synchronized mementos of the Event Log to an external *Choreographer*.

Figure 2 sketches a seven-step process to update the local Event Log of the repository and to create synchronized mementos at one or more archive services.

Step 1. The peer review service (the tilted square at the top of Fig. 2) sends an Announce Event Notification to the repository. This message informs the repository about a new peer review that is available pertaining to an artifact URL-A. This step is equivalent to Step 2 in Section IV. We skipped the implicit Step 1 of Section IV to avoid repetition in this scenario.

Step 2. The repository uses the information in the Announce message to create a new entry in a public Event Log URL-DN-E pertaining to artifact URL-A. The entry provides a proof that a review is available for the artifact URL-A. Further internal processing can be imagined, such as the creation of a (bi-directional) link on the landing page of the scholarly artifact pointing to the peer review at the peer review service.

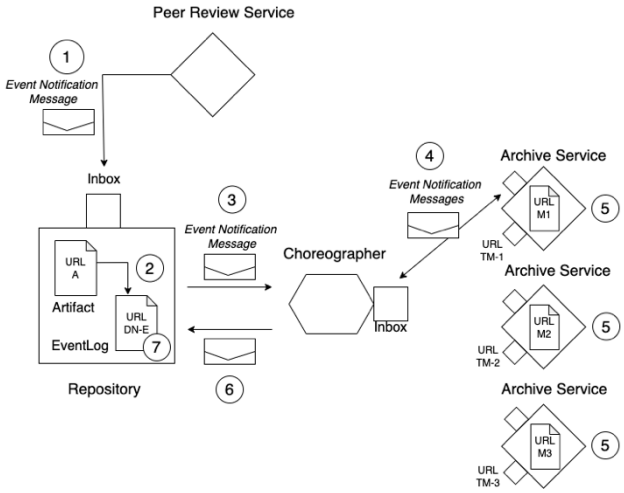

Figure 2. A repository receives an Event Notification message from a peer review service which is used to update a local Event Log. The repository outsources the creation of synchronized mementos of the Event Log to Choreographer. The Choreographer contacts one or more archive services to create mementos for the Event Log.

Step 3. The repository sends an Offer Event Notification message to the Choreographer with the location of the Event Log URL-DN-E. For the Event Log of the scholarly artifact a memento will be requested.

Step 4. The Choreographer receives the Offer Event Notification containing the location of the Event Log URL-DN-E. The Choreographer consults an internal database containing archive services that are best suited for creating mementos of the Event Logs. For instance, the database could contain a collection of archive services such as web archives, LOCKSS[7] networks, wikis, and git repositories. For each archive service, the Choreographer sends out an Offer Event Notification message to request the creation of a memento for the Event Log URL-DN-E.

Step 5. Each of the archive services receive the Offer Event Notification message and create a memento of the Event Log at URL-M1, URL-M21, and URL-M3, respectively. Each archive service sends an Announce Event Notification message to the Choreographer about the availability of a TimeMap for URL-DN-E in their archive.

Step 6. For each Announce message the Choreographer receives from archive services, a new Announce Event Notification message will be sent to the repository. These latter Announce messages inform the repository about new mementos that are available for the Event Log URL-DN-E.

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

Step 7. The repository uses the information in the incoming Announce Event Notification messages to set TimeMap links for the Event Log URL-DN-E Link HTTP header (if not yet available).

As a result of these seven steps, the following resources are created or updated, concerning the Event Log of the repository:

- A new Event Log entry at URL-DN-E containing information about a peer review value-adding service provided for the artifact at URL-A.
- Archived copies, mementos, of the Event Log URL-DN-E at URL-M1, URL-M2 and URL-M3 that are part of TimeMaps URL-TM1, URL-TM2 and URL-TM3.
- Link HTTP headers for URL-DN-E providing the location of the memento TimeMap of the archives.

VI. Event Notification Payloads

In this section we demonstrate the Event Notification messages required for a network containing:

- A repository at URL http://data.node which hosts a scholarly artifact at http://data.node/artifact (URL-A) with an associated Event Log at http://data.node/artifact/log (URL-DN-E). The repository has an LDN Inbox at http://data.node/inbox/ to receive Event Notification messages pertaining to the Event Log (and the scholarly artifact).
- A Choreographer at http://choreography.node with an LDN Inbox at http://choreography.node/inbox/ to receive Event Notification messages pertaining to the archiving value-added service it provides.
- An archive service at http://archive.node with an LDN Inbox at http://archive.node/inbox/ to receive Event Notification messages pertaining to the archiving value-added services it provides.
- A new memento for URL-DN-E at http://archive.node/memento/1 (URL-M1).
- A timemap for URL-DN-E at http://archive.node/map/http://data.node/artifact/log

Table 1 Overview of network nodes, resources and services used in Section III

Network nodes		
Repository	***Choreographer***	***Archive Service***
Network Role: Data node	*Network Role:* Service node	*Network Role:* Service node, TimeMap
Resources: - Artifact (URL-A) - Event Log (URL-DN-E)	*Resources:* (none)	*Resources:* - Memento URL-M - TimeMap URL-TM
Value-added Services: (none)	*Value-added Services:* - Relaying archive service request to a network of archive services	*Value-Added Services:* - Authentic memento versions of URL-DN-E

Table 1 provides an overview of the nodes that are introduced in this section with the resources and value-added services that are in focus for the use case in Section V.

The example Event Notification messages below follow the use case of Section V. The messages that are required for Section IV are similar.

Our examples below have a particular emphasis on the Event Logs rather than the scholarly artifact for which similar interactions with Choreographer and archive service are applicable.

From the repository to the choreography node

Listing 1 shows the Offer Event Notification message that is sent in step 3 of Section V from the repository to the Choreographer. This message offers the Event

Log at http://data.node/artifact/log to be archived in a trusted archive service.

Listing 1 Event Notification type "Offer" from the repository to the Choreographer.

```
{
   "@context":
     "https://www.w3.org/ns/activitystreams",
   "id": "urn:uuid:1",
   "type": "Offer",
   "actor": {
      "id": "http://data.node",
      "inbox": "http://data.node/inbox/",
      "type": "Application"
   },
   "object": {
      "id": "http://data.node/artifact/log",
      "type": "Document",
   },
   "target": {
      "id": "http://choreography.node",
      "inbox": "http://choreography/inbox/",
      "type": "Application"
   }
}
```

From the Choreographer to the archive service

Listing 2 shows the Offer Event Notification message that is sent in step 4 for Section V from the Choreographer to an archive service. The Event Log URL will be forwarded to the archive service as the subject of the new Offer message.

Listing 2 Event Notification type "Offer" sent by a choreography node to an archive service.

```
{
   "@context":
     "https://www.w3.org/ns/activitystreams",
   "id": "urn:uuid:2",
   "type": "Offer",
   "actor": {
      "id": "http://choreography.node/",
      "inbox":
        "https://choreography.node/inbox/",
      "type": "Application"
   },
   "object": {
      "id": "http://data.node/artifact/log",
      "type": "Document",
   },
   "target": {
      "id": "http://archive.node/",
      "inbox": "http://archive.node/inbox/",
      "type": "Application"
   }
}
```

From the archive service to the Choreographer

Listing 3 shows the Announce Event Notification message that is sent in step 5 of Section V from the archive service to the Choreographer. The message contains an `object` property which describes an update to the archive service's TimeMap. The new TimeMap entry contains the memento at http://archive.node/memento/1 for the Event Log at http://data.node/artifact/log. The message also contains the required `context` and `inReplyTo` fields that help the Choreographer to reconstruct the context of the previously sent offer.

Listing 3 Event Notification type "Announce" from an archive service to the Choreographer.

```
{
   "@context": [
     "https://www.w3.org/ns/activitystreams",
     { "iana": "https://www.iana.org/" }
   ],
   "id": "urn:uuid:3",
   "type": "Announce",
   "actor": {
      "id": "http://archive.node/",
      "inbox": "http://archive.node/inbox/",
      "type": "Application"
   },
   "context": "http://data.node/artifact/log",
   "inReplyTo": "urn:uuid:2",
   "object": {
       "id": "http://archive.node/map/
              http://data.node/artifact/log",
      "type": "Document",
      "iana:original":
        "http://data.node/artifact/log",
      "iana:memento":
       "http://archive.node/memento/1"
   },
   "target": {
       "id": "http://choreography.node/",
      "inbox":
         "http://choreography.node/inbox/",
      "type": "Application"
   }
}
```

From the Choreographer node to the repository

Listing 4 shows the Announce Event Notification message that is sent in step 6 of Section V from the archive service to the Choreographer. This message is almost a copy of the message in Table 3. The most important change is the `inReplyTo` field which contains the identifier of the original offer that was sent by the repository to the Choreographer. This information can be used by the repository to reconstruct the original context for which a service was requested.

Listing 4 Event Notification type "Announce" sent by the choreography node to the repository.

```
{
   "@context": [
     "https://www.w3.org/ns/activitystreams",
     { "iana": "https://www.iana.org/" }
   ],
```

```
    "id": "urn:uuid:4",
    "type": "Announce",
    "actor": {
       "id": "http://choreography.node/",
       "inbox":
          "https://choreographer.me/inbox/",
       "type": "Application"
    },
    "context": "http://data.node/artifact/log",
    "inReplyTo": "urn:uuid:1",
    "object": {
        "id": "http://archive.node/map/
                http://data.node/artifact/log",
       "type": "Document",
       "iana:original":
         "http://data.node/artifact/log",
       "iana:memento":
        "http://archive.node/memento/1"
    },
    "target": {
       "id": "http://data.node/",
       "inbox": "http://data.node/inbox/",
       "type": "Application"
    }
}
```

VII. TRUSTWORTHY EVENT LOGS

In our 2024 paper [14], we presented a proposal how the Event Log for a (scholarly) artifact URL-A could be discovered at the side of a data node (the repository) and at the side of a service node (the peer review service) by using Web Linking (RFC8288) [20].

At the side of a data node, a web agent can find the Event Log for artifact URL-A by issuing a HTTP GET or HEAD request against URL-A and follow the proposed link relation type `eventlog`:

```
HEAD http://data.node/artifact HTTP/1.1
Host: example.org
Accept: application/ld+json

HTTP/1.1 200 OK
Link: <http://data.node/artifact/log>;
      rel="eventlog"
```

At the side of a service node, a web agent can find the Event Log for the artifact URL-A by issuing a HTTP GET or HEAD request against the service node LDN Inbox which provides a Linked-Template HTTP header field [21]:

```
HEAD http://service.node/inbox/ HTTP/1.1
Host: example.org
Accept: application/ld+json

HTTP/1.1 200 OK
Link-Template: "/events/{artifact}";
      rel="eventlog"
```

The web agent uses the template and fills out URL-A for the `{artifact}` entry to get a link to a Event Log at the side of the service node containing value-added events for URL-A.

Using these two discovery techniques, the three subsections below sketch how the authenticity, the local completeness and global completeness of an Event Log can be verified.

Verify the authenticity of the Event Log

The process to verify the authenticity of the Event Log is sketched in figure 3 below.

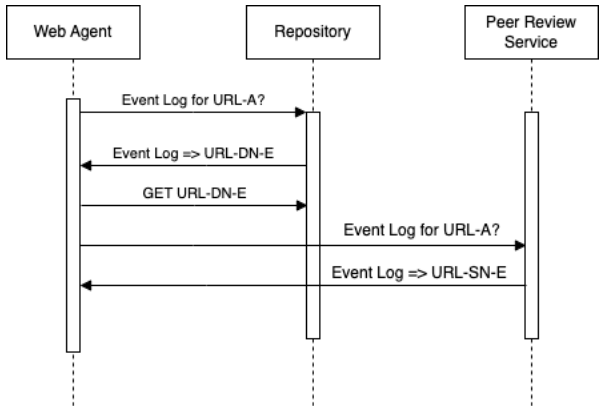

Figure 3 Process flow to verify the authenticity of Event Log entries.

Step 1. A web agent contacts the repository to request value-added events available for artifact URL-A by issuing an HTTP HEAD request.

Step 2. Using Web Linking, the repository points to the Event Log URL-DN-E.

Step 3. The web agent retrieves the Event Log URL-DN-E and discovers a peer review event originating from the peer review service which can be contacted at a particular LDN Inbox URL.

Step 4. Next, the web agent requests the location of an Event Log describing value-added events provided for URL-A from the peer review service. The web agent issues an HTTP HEAD request against the peer review service LDN Inbox.

Step 5. The peer review service responds with a HTTP Linked-Template header to be filled out by the web agent to find for URL-A the Event Log URL-SN-E.

Step 6. The web agent accesses the Event Log URL-SN-SE and compares the peer review entry with the peer review entry in URL-DN-E to verify the entry's authenticity. For instance, the web agent can compare the Event Notification messages available

in the Event Log of the peer review services with the stored versions in the Event Log of the repository.

Verify the local completeness of the Event Log

The process to verify local completeness of the Event Log is sketched in figure 4 below.

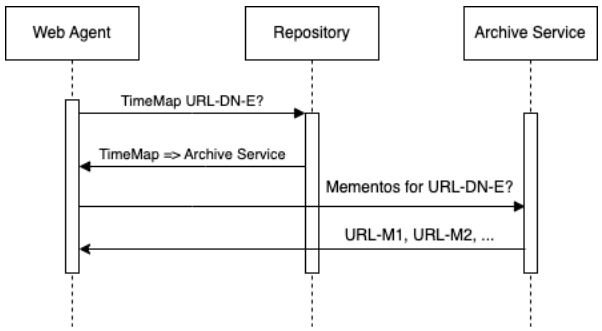

Figure 4 Process flow to verify the local completeness of Event Log entries.

In this process flow a web agent inspects whether the Event Log is genuinely append-only, with no entries being removed or edited.

Step 1. The web agent contacts the repository to request a trusted TimeMap for Event Log URL-DN-E by issuing an HTTP HEAD request against that URL.

Step 2. The repository points to the TimeMap at an archive service by including a `timemap` HTTP Link relation in the response:

```
HTTP/1.1 200 OK
Link: <http://archive.node/map/
http://data.node/artifact/log> ;
      rel="timemap"
```

Step 3. Next, the web agent uses the TimeMap URL to request memento copies for URL-DN-E from the archive service.

Step 4. The archive service responds with a list of all mementos for the Event Log URL-DN-E.

Step 5. The web agent compares these copies against the URL-DN-E version for any missing or edited past versions. Depending on the serialization of Event Logs data comparison techniques such as edit distances, RDF canonicalization[8] and checksums can be used to verify edits or deletions of past entries to the Event Logs.

Verify the global completeness of the Event Log

By utilizing the techniques above, one can verify the trustworthiness of the *local* version of an Event Log using a network of service providers. However, what remains unaddressed is the absence of data regarding value-added services that were not recorded in the Event Log but exist in the global network (e.g. due to some form of local censorship). Discovering censorship requires navigating the global web. Community efforts can be imagined decreasing the scope of this global search. For instance, national scholarly communities could mandate the use of trusted national archive services to ensure the authenticity of value-added services. Both Data nodes and service nodes will contribute to copies of archived information and inconsistencies can be discovered.

VIII. Discussion

Our requirements assume that archive services can create authentic mementos of Event Logs so that fixity information can be verified. Aturban et al. [24] show that in general cases, current web archives, such as the Internet Archive, routinely fail to offer authentic mementos to external applications when replaying archived web pages. The mementos that are presented to typical users of web archives are often not the raw data that was archived, but a processed version that presents the archive's best effort to create human interpretable past versions of the web. These 'replay' considerations would make the use of web archives for use-cases involving building trust in web resources less applicable. That being said, Aturban's concerns pertain to composite resources such HTML pages, for which the set of all resources required to recompose an archived page are not always available in the archive (not available for a particular timestamp), leading to temporal incoherence of the composite memento [25] or incomplete composite mementos [26]. However, since our scenarios only involve atomic textual resources, it is unlikely that replay issues will play any role when using web archives as archive services, especially not if the raw mementos of Event Logs are retrieved.

[8] https://www.w3.org/TR/rdf-canon/

Web archives are also not the only option for providing archive services. Public archives, serving authentic mementos, could be built on top of a variety of service providers such as git and wiki repositories that are memento-enabled using the Memento TimeGate software [22].

Our team is updating the Koreografeye software [23] to facilitate the services mentioned in this paper. We anticipate the timely completion of this effort to present a pilot demonstration to provide a tangible representation of our theoretical findings.

As a first attempt to serialize Event Logs, a serialization format, using fragmentation techniques such as Linked Data Event Streams (LDES) [26], will be used. LDES provides a logging format for RDF messages, such as Event Notifications, with the capability to calculate fixity information (checksums) for each stable fragment. To provide trust by a third party, each of these checksums can be compared at the side of the archive service by inspecting the checksum of the memento of the Event Log.

IX. Conclusion

This article examines ways to enhance current decentralization initiatives, such as COAR Notify, to create an alternative scholarly communication system. It suggests that publishing public Event Logs can enhance transparency within the scholarly communication process. To provide a level of trust in such networks, we advocate for a network of trusted archive services that provide authentic mementos of Event Logs and in general for all scholarly artifacts. Our findings in this paper will provide a guide for standardization efforts for the Event Logs themselves, and implementations demonstrating trust in decentralized networks. We hope that the results of our project will bring a truly ,decentralized transparent, and trustworthy scholarly communication infrastructure closer to reality.

References

[1] Fraser, N., Brierley, L., Dey, G., Polka, J., Pálfy, M., Nanni, F., Coates, J. (2021). The evolving role of preprints in the dissemination of COVID-19 research and their impact on the science communication landscape. PLoS Biol 19(4): e3000959. https://doi.org/10.1371/journal.pbio.30009591

[2] Majumber, M., Mandl, K. (2020). Early in the epidemic: impact of preprints on global discourse about COVID-19 transmissibility. The Lancet 8(5): E627-E630. https://doi.org/10.1016/S2214-109X(20)30113-31

[3] Escamilla, E., Klein, M., Cooper, T., Rampin, V., Weigle, M.C., Nelson, M.L. (2022). The Rise of GitHub in Scholarly Publications. In: Silvello, G., et al. Linking Theory and Practice of Digital Libraries. TPDL 2022. Lecture Notes in Computer Science, vol 13541. Springer, Cham. https://doi.org/10.1007/978-3-031-16802-4_15

[4] Escamilla, E., Salsabil, L., Klein, M., Wu, J., Weigle, M.C., Nelson, M.L. (2023). It's Not Just GitHub: Identifying Data and Software Sources Included in Publications. In: Alonso, O., Cousijn, H., Silvello, G., Marrero, M., Teixeira Lopes, C., Marchesin, S. (eds) Linking Theory and Practice of Digital Libraries. TPDL 2023. Lecture Notes in Computer Science, vol 14241. Springer, Cham. https://doi.org/10.1007/978-3-031-43849-3_17

[5] Candela, L., Castelli, D., Manghi, P. and Tani, A. (2015), Data Journals: A Survey. J Assn Inf Sci Tec, 66: 1747-1762. https://doi.org/10.1002/asi.23358

[6] San Francisco Declaration on Research Assessment. Retrieved March 15, 2024 from https://sfdora.org/read/

[7] COAR Notify Initiative. Retrieved March 15, 2024 from https://www.coar-repositories.org/notify/

[8] Hanson, M., Barreiro, P., Crosetto, P., Brockington, D. (2023). The strain on scientific publishing. arXiv:2309.15884. https://doi.org/10.48550/arXiv.2309.15884

[9] Avissar-Whiting, M., Belliard, F., Bertozzi, SM., Brand, A., Brown, K., Clément-Stoneham, G., et al. (2024) Recommendations for accelerating open preprint peer review to improve the culture of science. PLoS Biol 22(2): e3002502. https://doi.org/10.1371/journal.pbio.3002502

[10] Schulz, R., Barnett, A., Bernard, R. et al. Is the future of peer review automated?. BMC Res Notes 15, 203 (2022). https://doi.org/10.1186/s13104-022-06080-6

[11] Hochstenbach, P., Van de Sompel, H., Vander Sande, M., Dedecker, R., Verborgh, R. (2022). Event Notifications in Value-Adding Networks. In: Silvello, G., et al. Linking Theory and Practice of Digital Libraries. TPDL 2022. Lecture Notes in

Computer Science, vol 13541. Springer, Cham. https://doi.org/10.1007/978-3-031-16802-4_11

[12] Hochstenbach, P., Vander Sande, M., Dedecker, R., Walk, P., Klein, M., Van de Sompel, H. Event Notifications in Value-Adding Networks. Retreived March 15, 2024 from https://www.eventnotifications.net

[13] Implementation of the COAR Notify protocol in DSpace 8. Retrieved March 15, 2024 from https://wiki.lyrasis.org/display/DSPACE/Implementation+of+the+COAR+Notify+protocol+in+DSpace+8

[14] Hochstenbach, P., Verborgh, R., Van de Sompel, H. (2024). Event Notifications and Event Logs: Transparent Sharing of Artifact Life Cycle Data. IDCC 2024. Accepted as IJDC Conference paper

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

[16] Cabanac, G., Oikonomidi, T. & Boutron, I (2021). Day-to-day discovery of preprint–publication links. Scientometrics 126, 5285–5304 https://doi.org/10.1007/s11192-021-03900-7

[17] Abalkina, A. (2021). Publication and collaboration anomalies in academic papers originating from a paper mill: evidence from a Russia-based paper mill. arXiv:2112.13322. https://doi.org/10.48550/arXiv.2112.13322

[18] Prewett, K., Prescott, G. (2020). Blockchain adoption is inevitable—Barriers and risks remain. The Journal of Corporate Accounting & Finance 31(2) https://doi.org/10.1002/jcaf.22415

[19] Van de Sompel, H., Nelson, M., Sanderson, R. (2013). HTTP Framework for Time-Based Access to Resource States – Memento (RFC 7089). IETF https://www.rfc-editor.org/rfc/rfc7089

[20] Nottingham, M. (2017). Web Linking (RFC 8288).IETF https://www.rfc-editor.org/rfc/rfc8288

[21] Nottingham, M. (2024). The Link-Template HTTP Header Field. IETF https://www.ietf.org/archive/id/draft-ietf-httpapi-link-template-03.txt

[22] Shankar, H. timegate https://github.com/mementoweb/timegate

[23] Hochstenbach, P. Koreografeye (Version 0.4.9) https://github.com/eyereasoner/Koreografeye

[24] Aturban, M., Klein, M., Van de Sompel, H., Alam, S., Nelson, M., Weigle, M. (2023). Hashes are not suitable to verify fixity of the public archived web. PLoS ONE 18(6): e0286879. https://doi.org/10.1371/journal.pone.0286879

[25] Ainsworth, S., Nelson, M., Van de Sompel, H. (2014). A Framework for Evaluation of Composite Memento Temporal Coherence. arXiv:1402.0928.https://doi.org/10.48550/arXiv.1402.0928

[26] Ainsworth, S., Nelson, M., Van de Sompel, H. (2015). Only One Out of Five Archived Web Pages Existed as Presented. In Proceedings of the 26th ACM Conference on Hypertext & Social Media (HT '15). Association for Computing Machinery, New York, NY, USA, 257–266. https://doi.org/10.1145/2700171.2791044

[27] Colpaert, P. Linked Data Event Streams. Retrieved March 15, 2024 from https://semiceu.github.io/LinkedDataEventStreams/

ACKNOWLEDGMENTS

This work is funded by SolidLab Vlaanderen (Flemish Government, EWI, and RRF project VV023/10). The authors would like to thank COAR Notify (Kathleen Shearer, Paul Walk) for involving them from the outset, allowing them to provide input and gain valuable insights in the scholarly communication process.