Preserving the Archival Bond in Distributed Ledgers: A Data Model and Syntax

Victoria L. Lemieux
The University of British Columbia
481-1961 East Mall
Vancouver, British Columbia, Canada
1-604-822-2404
vlemieux@mail.ubc.ca

Manu Sporny
Digital Bazaar
203 Roannoke St. West
Blacksburg, Virginia, USA
1-540-961-4469
mssporny@digitalbazaar.com

ABSTRACT
Distributed cryptographic ledgers, such as the blockchain, are now being used in recordkeeping. However, they lack a key feature of more traditional recordkeeping systems needed to establish the authenticity of records and enable reliance on them for trustworthy recordkeeping. The missing feature is known in archival science as the archival bond – the mutual relationship that exists among documents by virtue of the actions in which they participate. In this paper, we propose a novel data model and syntax using core web principles that can be used to address this shortcoming in distributed ledgers as recordkeeping systems.

Categories and Subject Descriptors
• ***Information systems~Semantic web description languages*** • *Information systems~Ontologies* • ***Computer systems organization~Peer-to-peer architectures***

General Terms
Reliability, Security, Legal Aspects, Verification.

Keywords
Distributed legers; archival science; archival bond; trust; evidence.

1. INTRODUCTION
Blockchain technology is a specific implementation of distributed cryptographic ledgers often described as providing a decentralized and continuously growing immutable record of transactions. As such, it is a recordkeeping technology, in the archival science sense of the term; that is, it is concerned with the keeping of records, which can be described as documents (an indivisible unit of information, not necessarily in paper form, such as a ledger entry) made or received in the course of practical activity and set aside for future action or reference [1]. Another way of looking at the record is as a type of information system that provides "persistent representations of activities or other occurrents [i.e. actions], created by participants or observers of those occurrents or by their proxies; or sets of such representations representing particular occurrents" [2] [3].

From the days of early records on clay tablets, societies have relied upon them as evidence of rights and entitlements. The systems for creating and keeping records – recordkeeping systems – have therefore had to ensure that records could be trusted to provide reliable and authentic evidence. Quite frequently, such systems have failed to achieve this important goal because they could be tampered with or otherwise altered, or the records in them could be lost. This is never more evident than in digital recordkeeping systems in which records are volatile and subject to loss, intentional or unintentional alteration, contamination, or corruption and where their authorship, provenance, or chain of custody may be difficult or impossible to determine [4].

Blockchain-based recordkeeping systems have been advanced as a solution to the inherent weaknesses of more traditional recordkeeping systems as, in theory, distributed ledgers create immutable records protected from the usual vicissitudes of digital records through cryptographically-enabled validation processes and a decentralized architecture.

While blockchain-based recordkeeping in theory addresses the problem of mutability of records, improving upon some of the weaknesses of contemporary information systems, in practice the design of blockchain-based systems for application in recordkeeping still has a number of flaws. This paper focuses on one such flaw: the absence of a means to instantiate and preserve the *"archival bond*". **The archival bond expresses the network of relationships that each record has with the records resulting from the same activity** [1] [4]**.** Instantiating and maintaining the bond is essential to ensuring the continuing authenticity of records, a key feature of trustworthy recordkeeping [4].

In the following section we elaborate on the nature of the archival bond and explain why it is of utmost importance in preservation and use of records as evidence. We then explain the nature of the gaps in current blockchain-based recordkeeping systems in relation to the archival bond, and in the final section, we describe our proposed approach to redress this important gap.

2. THE ARCHIVAL BOND
2.1 Records as Documentary Evidence
In archival theory, closely linked to legal theory, there are three characteristics that contribute to the trustworthiness of records: accuracy, reliability and authenticity [4] [7] [8] [9] [11]. For records "set aside" in a blockchain-based recordkeeping system,

WWW 2017, April 3-7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
DOI: http://dx.doi.org/10.1145/3041021.3053896

authenticity may well be the most critical of the three concepts, since accuracy and reliability of records will often be determined before the records are "anchored" in a blockchain.

The Society of American Archivists defines authenticity as: "The quality of being genuine, not a counterfeit, and free from tampering, and is typically inferred from internal and external evidence, including [a record's] physical characteristics, structure, content, and context" [6]. Authenticity does not automatically imply reliability of the content of the record [4] [6] [7]. ISO 15489, the international records management standard, identifies authenticity as follows: "An authentic record is one that can be proven: a) to be what it purports to be, b) to have been created or sent by the person purported to have created or sent it, and c) to have been created or sent at the time purported" [10]. **Implicit in the definition of authenticity is the notion that records have a unique identity, for without such it would be impossible to establish that the record is what it purports to be.** In other words, it would be impossible to prove that a records was an inauthentic copy of another record (i.e., a forgery), unless both records (the record to be proven and the record that serves as proof) have unique identities.

If records are inauthentic they cannot serve as evidence (except as evidence in relation to their own inauthenticity), and therefore important rights and entitlements cannot be upheld.

2.2 The Role of the Archival Bond in Establishing Authenticity

The Italian archivist Giorgio Cencetti has defined the archival bond as the "originary, necessary and determined" relationship between and among records that participate in the same activity" [12]. Duranti writes that, "Ultimately, the key to the existence of an electronic record is the archival bond . . . Differently from the context, the archival bond is not external to the record, but an integral part of it" [10]. An example is offered by two documents that simply read: "press the red button." In one case, the document is linked to other documents through an archival bond that establishes them all as part of an elevator or lift repair process. In another case, they are linked by an archival bond that establishes them as part of a nuclear launch process. The content of the documents is the same, and even their bit structure may be the same, but the identities of the documents **as records** (i.e., evidence of facts about acts or transactions) are completely different by virtue of the different procedures of which they form a part (as represented by the archival bond). In the case of digital records, it would be impossible to prove that a record was an authentic representation (i.e., a copy) of another record, unless both items (the one to be proven authentic and the one that was reproduced) have unique identities. In other words, it is not sufficient to refer to or match the content of the records, or even their bit structures, in order to establish their authenticity; the archival bond must be made explicit and interpretable in order to ascertain the unique identity of each document as a record of the procedurally bound facts contained within it.

The archival bond contains within itself the direction of the cause-effect relationship of the procedure which gives rise to records, and it is therefore the primary expression of the **development** of the activity in which the document participates, rather than just facts about the act that the document embodies (e.g. property sale, securities trade, or degree conferment) [10] [13]. For example, in a Bitcoin transaction, the cause effect relationship is represented very simply by Bitcoin moving from a wallet at address A to a wallet at address B. In more complex transactions, however, there are many steps, and each of these steps may generate its own representation (i.e., record) to memorialize the development and facts about of the activity. In a real estate transaction, for example, a seller will first put a property on the market. Depending on legal jurisdiction, the document representing this first step in a real estate transaction may be a listing agreement with a real estate agent. Next there may be a bank appraisal document, which assesses the value of the property. There may also be advertisements in various forms, such as newspaper ads or online listings. Finally, there will be a contract for sale document. It is not uncommon to see that various versions of the contract document are made as the contract is passed back and forth between the seller (or agent) and the buyer (or agent) while counter offers are made. This is done until such time as a final sale price is agreed. At this point, both parties and their agents sign the contract and once the sale is recorded in the official land titles register, ownership of the property is officially transferred and the real estate transaction is complete. All of the documents created as a part of completing this transaction are procedurally bound together; that is, they all share an archival bond to a specific real estate transaction and to one another.

It is this natural mutual relationship between documents that participate in the same causal sequence that defines the archival bond. **This mutual relationship also determines the unique meaning and identity of a record, and because a record must have a unique identity before its authenticity can be determined, it provides a foundation for establishing authenticity.**

Even though the term archival bond is used in singular form in this paper and in much of the archival literature, in reality a document may have a multitude of relationships with other documents. Thus a single document may have many archival bonds forming a dense network of relationships. The totality of these bonds or relationships is what gives a particular documentary object its unique identity as a record and permits one to differentiate it from another (e.g., for the purposes of detecting a forgery or understanding the significance of the document in terms of the content it conveys). If one of these bonds is changed by breaking the procedural link, the nature of the record is altered. This is clearly illustrated by our example: the implication of each document is quite different, even though the content of each ("press the red button") is exactly the same. We would not understand the implication of one document versus another without linking each document back to its procedural context.

This network of relationships is seldom static over time and space: as McNeil notes, acts of continuous and discontinuous change transform the meaning and authenticity of records as they are transmitted, a process which she calls *Archivalterity* [14]. This is why some archival theorists write of the record as "always becoming" [15] The tracing of these changes is encompassed in the analysis of provenance, which is part of the forensic assessment of the authenticity of records.

2.3 Traditional Mechanisms of Rendering and Preserving the Archival Bond

In paper-based recordkeeping systems, rendering the archival bond and preserving it was relatively straightforward. This was generally achieved through the assignment of a classificatory code, usually during the process of document registration, based on function and/or activity that linked the document as record (via the code) to other records participating in the same function and activity. This link was manifested as physical proximity in paper-based systems, for example, by placing the document in the same file folder as the records to which it was procedurally bound [4].

In the digital world, physical proximity has been replaced by establishment of a logical link between the document as a digital object and other documents participating in the same action. This logical link is instantiated by metadata that represent the functional classification code. These essential mechanisms enable the fixing of the record's identity and authenticity, and subsequent preservation and assessment of the authenticity of the records [4].

Currently, there is no similar mechanism in blockchain-based recordkeeping systems to establish the archival bond **while at the same time preserving the unique identity of each record**, as we explain in more detail in the following section.

3. BLOCKCHAIN-BASED RECORDKEEPING AND THE ABSENCE OF A MECHANISM TO ESTABLISH THE ARCHIVAL BOND

We illustrate the issue with an overview of how Bitcoin creates and updates a distributed public ledger. The process starts when Bitcoin address A proposes the transfer of Bitcoin to another address B. Next the Bitcoin distributed "mesh" network checks the public ledger that sufficient Bitcoin exists in the wallet at address A. If there is sufficient Bitcoin, specialized nodes called miners will bundle the proposal with other reputable data representing transactions to create a new block for the Blockchain. Here it is important to note that the bundling of reputable transactions into blocks is completely agnostic as to the nature of those representations of transactions (i.e., they can relate to any transaction of any type from any source in a public blockchain like Bitcoin). The blocks are cryptographically "hashed"; that is, they are used as input to an algorithm that converts them into a fixed-size alphanumeric string, which is called the hash value (sometimes also called a message digest, a digital fingerprint, a digest or a checksum). That hash is put, along with some other data (e.g., a nonce), into the header of the proposed block. This header then becomes the basis for the "proof of work" performed by the miner nodes on the Bitcoin network. When a miner node arrives at a solution to the proof of work, other nodes check it and then each node that confirms the solution updates the Blockchain with the hash of the header of the proposed block. This becomes the new block's identifying string, now part of the distributed ledger in the Blockchain. Address A's payment to address B, and all the other transactions the block contains, are confirmed [16].

It is a common mistake to think that because every block of transactions (and thereby every transaction) in a proof-of-work blockchain is transitively bonded to every previous block, by virtue of the way proof of work functions, that the archival bond is preserved. However, even though the time-ordered nature of the transactional records is preserved, the link to their procedural context, and relationship to other transactional records relating to the same procedure, is not. For example, let us say that the payment described above concerned a series of payments specified under the terms of a particular smart contract. These payments would then have an archival bond, being naturally related to one another by virtue of their procedural association. Since the formation of blocks as described above is agnostic to this feature of the records, with blocks forming not on the basis of shared procedural origins but rather on the basis of time, the archival bond is not rendered explicit and information needed to establish the unique identity and authenticity of the transactional records may be all but lost.

Some blockchain-based recordkeeping solutions[1] take a slightly different approach to that described in the example above in that they hash all the documents that are part of the logical transaction (i.e., the same action) and place all the hashes into a metadocument, which is then hashed again. The latter hash is then the item that is placed into the blockchain. While this does establish and preserve the archival bond, in that it immutably links together documents taking part in the same action, it has several shortcomings from a recordkeeping perspective. First, and most critically, it fails to preserve the unique identity of each transactional record comprising the metadocument. While it is true that the archival bond between the documents is established and preserved in this approach, the hashing of the metadocument transforms the escapsulated hashes into a new document which destroys (since the hash cannot be reverse engineered) the individual identities of the documents within the metadocument that have contributed to the formation of the new hash. As a result, subsequent determination of the authenticity of all those documents that contributed to the formation of the metadocument becomes impossible.

The other issues are more practical. While aggregation of the documents into the metadocument is efficient from the standpoint of information processing and addressing blockchain size constraints, the downside is that, in order to establish the archival bond, it forces one to wait to bundle all logically related transactions together into the metadocument before hashing and anchoring in the blockchain. In real-world recordkeeping, however, actions often take place in time-ordered sequences that can span a considerable amount of time. For example, a financial derivative contract may stipulate several payments over a number of years at key "trigger" points. To instantiate and retain the archival bond between these payments using the above method would require years of waiting in order to anchor the transactions into the blockchain. In addition, retrieval of the individual transaction records would be problematic as a result of bundling them into the metadocument and generating a new hash. Finally, this approach does not address the need to establish an archival bond between documents recorded on different ledgers or between related documents, some of which are on chain and others off chain. So, clearly, the approach would not work for every use case, although it may well be appropriate for some. Thus, an alternate solution must be sought.

An alternative approach might be to use transaction metadata (e.g. the OP_RETURN field in Bitcoin) to establish an archival bond between transactions in a blockchain. The use of OP_RETURN is common to a number of blockchain-based solutions, and, indeed, is on the rise according to some sources [17]. To illustrate how this approach could work on a Bitcoin blockchain, in a manner similar to the addition of a descriptor to a wire transfer, OP_RETURN script opcode could be used to mark a transaction with procedural metadata (e.g., a classificatory code). Setting aside debates about whether the Bitcoin blockchain should contain any data not necessary to validate a Bitcoin transaction, the primary difficulty is that the OP-RETURN data does not form part of the Bitcoin transaction per se and thus is not validated in the same way. It could thus be altered or severed from the Bitcoin transaction record to which it relates. OP-RETURN space limitations also could potentially pose a problem. Currently, the default Bitcoin client relays OP_RETURN transactions up to 80 bytes [17], which is sufficient space to incorporate metadata needed to instantiate the

[1] For example, Tierion (https://tierion.com) and Stampery (https://stampery.com).

archival bond but may be insufficient if that metadata must be added to other pieces of metadata serving other functions. This is not a universal problem, since different blockchains have different OP_RETURN space constraints; however, it is a potentially limiting feature of this approach.

4. PRESERVING THE ARCHIVAL BOND IN BLOCKCHAIN-BASED RECORDKEEPING USING LINKED DATA

In this section we propose a mechanism for rendering the archival bond in blockchain-based recordkeeping systems explicit by using an extensible, protocol-agnostic data model and syntax for expressing a set of ordered events in a decentralized system in a way that can be cryptographically verified [18]. The model takes advantage of existing web principles and standards, so that the archival bond can be determined independent of any particular application. This approach is useful when recording events, such as financial transactions, transfer of property, or time-stamped data that must be shared among participating parties. A primary goal of this ledger data model and format is flexibility, allowing for "pluggability" of consensus algorithms, data structures, and the type of data that can be stored in the ledger. The standardized data model and syntax allows for events, such as storage and retrieval, to occur independent of a particular implementation or application.

4.1 Data Model

To provide the greatest amount of flexibility, the data model is a graph, which aligns well with the network of relationships created by the archival bond between records [3] [19]. If a formalized data model is desired, RDF may be utilized but is not necessary. To ensure that the syntax will be easily adopted by developers, and to support a graph-based data model, it is recommended that the implementation syntax is JSON-LD (or similarly compatible syntax). Other RDF-compatible syntaxes may also be used to express the ledger semantics. This ensures that the syntax can be processed by widely adopted JSON tooling while also ensuring that the data model is robust enough to handle decentralized extensibility without name clashes or conflicts.

A ledger consists of a series of entries. This section outlines the basic types of entries that all web ledgers support. A **configuration event** specifies which software algorithms should be applied when processing a particular web ledger. The first entry in a ledger is typically an entry called a **genesis event** (aka seed event) and typically contains the configuration event. The basic structure of a configuration event, expressed in JSON-LD, is provided below:

```
{
 "@context": "https://w3id.org/flex/v1",
 "id": EVENT_ID,
 "type": "LedgerConfigurationEvent",
 "ledgerConfig": {
  "id": LEDGER_ID,
  "type": "LedgerConfiguration",
  "name": "example",
  "description": "This is an example ledger.",
  "storageMechanism": STORAGE_DATA_STRUCTURE,
  "consensusAlgorithm": CONSENSUS_ALGORITHM,
 "previousEvent": {
  "hash": "urn:sha256:00000000 ... 00000000"
 },
 "signature": SIGNATURE_VALUE
}
```

The **configuration event** provides the rules that will be used to determine the integrity of the blockchain. In the **configuration event** provided in the previous example, LEDGER_ID is used to uniquely identify the ledger (e.g. did:f6ea280f-8011-4502-a29f-464954de3427). EVENT_ID is used to uniquely identify the event in the ledger (e.g. did:f6ea280f-8011-4502-a29f-464954de3427/events/1. CONSENSUS_ALGORITHM is used to provide the type and parameters for the algorithm that will determine when consensus has been reached (e.g. Proof of Work, M-of-N Signatures, 1-of-N Signatures, Proof of Stake, etc.) STORAGE_DATA_STRUCTURE is used to identify the storage mechanism that is used in the ledger (e.g. SequentialList, MerkleTree, etc.) so that serializing and deserializing the contents of the ledger remains consistent across ledgers. SIGNATURE_VALUE is used to perform the cryptographic proof that the ledger entry was created by the entity identified in the signature.

A standard configuration entry, which can appear in any subsequent block, only differs from a genesis event in that the *previousEvent* value refers to the event before the current event. The **PREVIOUS_EVENT_ID** is the identifier of the previous event in the ledger (e.g. did:f6ea280f-8011-4502-a29f-464954de3427/events/1) and the **PREVIOUS_EVENT_HASH** value (e.g. urn:sha256:abd465d34f7a3f0f7d849550eb9fc32c17d12881da6b524da0a96e12cc984538 is a hash of the previous event in the ledger.

A **storage event** stores data in a ledger by specifying a list of UPDATE_OBJECTS. The basic structure of a storage event, expressed in JSON-LD, is provided below:

```
{
 "@context": [
  "https://w3id.org/flex/v1",
   MARKET_VERTICAL_CONTEXT],
 "id": EVENT_ID,
 "type": "LedgerStorageEvent",
 "replacesObject": [ UPDATE_OBJECTS ],
 "previousEvent": {
  "id": PREVIOUS_EVENT_ID
  "hash": PREVIOUS_EVENT_HASH
 },
 "signature": SIGNATURE_VALUE
}
```

When storing data in the ledger, one may use a **MARKET_VERTICAL_CONTEXT** to add market-specific

semantics to the updated objects (e.g. https://w3id.org/vaccinations/v1). It is expected that each object in UPDATE_OBJECTS has a unique identifier to enable global uniqueness and searching based on that identifier. A **consensus event** is an assertion that there is agreement on a subset of entries in a ledger. Some ledgers do not require consensus events as each event establishes an acceptable level of consensus. Other ledgers require consensus events after a pre-determined amount of time (e.g. every 1,000 events). We note that how often a consensus event should occur is highly dependent on the use case being addressed. A consensus event has a relatively simple definition. It consists of a number of **SIGNATURES** from notaries as determined by the previous configuration event. Finally, a **checkpoint event** may be used to quickly bootstrap new mirrors for a ledger such that the entire history of the ledger need not be downloaded and replayed for a node to become operational.

4.2 HTTP API

The previous section defined the data model and messages that can be used to create and operate a ledger. This section provides the HTTP API endpoints, still a work in progress, that may be used in conjunction with the messages in the previous section to create and operate a ledger. A ledger is created by performing an HTTP POST of a *LedgerConfigurationEvent* to the *ledgerCreateService*. One can get a list of active ledgers on a server by performing an HTTP GET on the *ledgerListService*. The metadata related to a ledger may be fetched by performing an HTTP GET on the *ledgerMetadataService*. Appending to a ledger can be achieved by performing an HTTP POST of a *LedgerStorageEvent* to the *ledgerAppendService*. A ledger read for an event is achieved by performing an HTTP GET on a ledger event identifier. The list of ledger events is available by performing an HTTP GET on the *ledgerIndexService*, and the current state machine of a ledger may be queried by performing an HTTP GET on the l*edgerQueryService*.

4.3 Explanatory Example

The following presents an example of an update to storage on the ledger relating to a real estate transaction involving the transfer of ownership of a property, just as in the example in section 2:


```
{
 "@context": [
  "https://w3id.org/flex/v1",
  "https://w3id.org/housing/v1",
 ],
 "id": "https://vhda.va.us.gov/ledgers/webville/houses/2",
 "type": "LedgerStorageEvent",
 "previousEvent": "https://example-consortium.com/private-ledgers/loans/real-estate/1",
 "replacesObject": [{
  "id": "https://vhda.va.us.gov/properties/3829344",
  "propertyAddress": {
   "street": "263 Main Street",
   "locality": "Webville",
   "region": "VA",
   "postalCode": "24736-3726",
   "country": "US"
  },
  "owner": {
   "name": "Jane Smith",
   "postalAddress": { ... }
 }],
 "signature": {
  "type": "LinkedDataSignature2016",
  "created": "2016-02-22T02:10:21Z",
  "creator": "https://webville.va.us.gov/i/planning-department/keys/1",
  "signatureValue": "cNJGLFqT/d/90D4GFzv...yKPiw=="
 }
}
```


We now describe the above elements and discuss how they may be used as a mechanism to render the archival bond.

@context establishes the context and how to read and interpret the blockchain by fetching a machine readable ontology (i.e., the ontology for the ledger itself as well as real estate transactions in this case). The purpose of the ontology is to establish the context, including functional and procedural, of the transaction, which is a necessary precondition to render the archival bond among related records linked together by their participation in the same action. This ontology would ideally be created by domain experts in the area following specific procedures [3] [20]. The ledger data model and syntax make no assumption about which ontology is used. Ontologies can also be layered to enrich the expression of context. It is also possible to switch ontologies from block to block and object to object and to have an array of objects in which ontologies are switched for each object. **It is by means of this mechanism that the archival bond can be established, since the entry can be linked by the ontology to the procedural action of which it forms a part in order to establish the record's identity (**in this example – a real estate transaction) as well as being grouped into semantically meaningful classes for purposes of interpretation and retrieval.

The following line denoted by **id** is the identifier of the block, which can be any URI scheme (e.g. URL, URN, IS-DFS, etc.).

The type field refers to the type of event, i.e., a storage event.

The **previousEvent** field specifies the previous event for the block.

The **replacesObject** field is replacing the current object in the space, if one exists, with the new object. If a previous object does not exist with the same identifier, a new object is created. The list of objects is used to update the state machine associated with the ledger. ID 3829344 refers to a particular property. The address data is pulling from a property database in this example, but it could be reading directly from the blockchain. While the property address may not be updated after it has been established, the owner name is expected to change on a more frequent basis. Rules should specify

how the entry is updated i.e., the current address should match the address of the previous transaction).

The final element is the signature of the person or entity that committed the block. Validation of the signature is determined by the specific implementation, and the signature can be validated in different ways. This is achieved by having the configuration block tell the ledger what rules should be used (e.g. this is a permissioned ledger and only specific public keys can write to the ledger; use a signature scheme where three to five signatures are required to establish the validity of an event).

This above described example represents the last step in the real estate transfer procedure discussed in section 2. Just as in that illustrative example, there may be many other steps in the procedure (i.e., real estate transaction). If each of these were recorded to a distributed ledger in the usual manner described in section 3 (i.e., like a simple transfer of Bitcoin from one wallet to another), the link (archival bond) between the procedural context that led to creation of the documents would be lost, their unique identities as records of a specific real estate transaction could not be established and, thus, it would be very difficult to validate their authenticity. Using the approach outlined above, however, it is possible to link all of these documents to their procedural context and to one another to instantiate the archival bond, establish the unique identities of the documents as records, and, ultimately, to support the determination of their authenticity.

4.4 End User View

The Figures below illustrate the process of adding data to a distributed ledger, including the instantiation of the archival bond, from the perspective of the end user. Figure 1 presents a ledger data entry interface, pre-populated with industry-specific use case data (in this example, a use case involving classification of financial flows on a watch list).

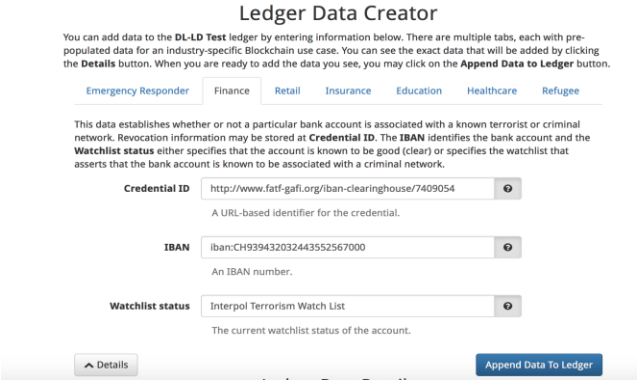

Figure 1. Ledger data entry screen [21].

Figure 2 shows the unique ID ("did") for the storage event entry in the ledger DL-LD Test. The text below the unique ID presents the details of what is represented by the transaction that was recorded in the ledger.

4.5 Relationship to Other Blockchains

The intention is that this data model and ontology should be used in concert with existing blockchain systems or in new Linked Data Blockchain systems.

When used with existing blockchain systems, hashes of the entire state of the system can be taken and injected into existing blockchains like Bitcoin, Ethereum, Hyperledger, or Corda using Blockchain receipt techniques like Chainpoint [22].

When used with new Linked Data Blockchain systems, the data model and ontology can be used directly to implement the core expression of the blockchain, as it is in the Flex Ledger prototype [21].

Hybrid models also exist, such as combining the EthOn[20] ontology with the one proposed in this document. In theory, the two data models should be compatible but further research is necessary to ensure that an implementation of a hybrid model results in an operational system.

There are many challenges when it comes to moving the proposed system in the direction of a global standard. Some of these challenges include: 1) a reluctance from blockchain implementers to work on standards, 2) a complete ontology for Bitcoin, Corda, and Hyperledger and mapping those systems onto this ontology, 3) ensuring that the ontologies for storage algorithms and consensus algorithms are complete enough to result in workable solutions, and 4) the difficulty of having a network as large as Bitcoin or Ethereum ensure that the data model, security model, and incentives scale to thousands of nodes participating.

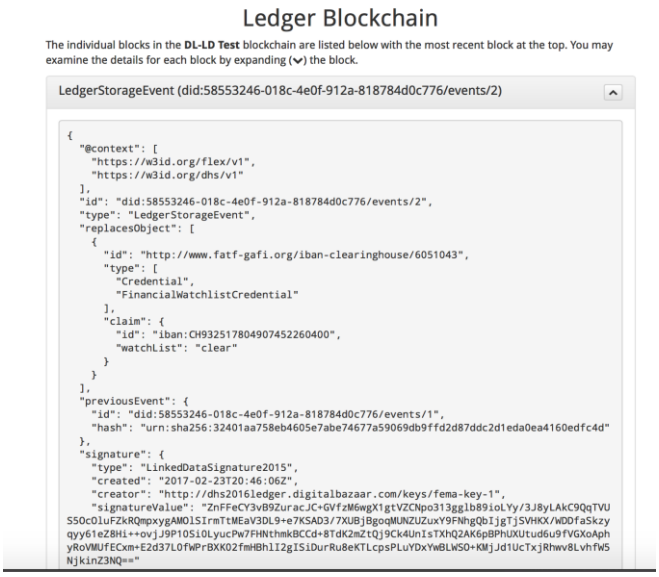

Figure 2. Example ledger entry [21].

5. CONCLUSION

This paper has discussed how one challenge of creating trustworthy records used as evidence of transactions can be addressed in blockchain-based recordkeeping systems through the use of core web principles and standards, such as URIs, HTTP, and JSON-LD, to instantiate an important principle of trustworthy recordkeeping – the archival bond. We have discussed a data model and syntax for expressing a set of ordered events in a decentralized system in way that can be cryptographically verified as a mechanism to establish the bond and explained that, through use of ontologies to represent the procedural context of ledger entries, it is possible to instantiate the archival bond between ledger entries as records of a variety of transactions. We propose that this supports establishment of the unique identity of each transaction, contributing to its creation as an authentic record and supporting later assessment of its authenticity. This capability is a critical precondition for reliance on ledger

entries as trustworthy records of transactions, and use of distributed cryptographic ledgers in recordkeeping. In addition, the approach not only establishes the archival bond between individual entries on the same ledger, **but also enables the establishment of the bond between entries across different ledgers**. Our work on this method is still ongoing and at the experimental stage, and there remain open questions regarding interoperability and assurance of the persistence of the bond; however, we think that using the linked data approach described above holds promise to address a key gap in existing blockchain-based recordkeeping system design.

6. ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers who provided feedback on an earlier draft of this paper. Any errors and omissions remain the sole responsibility of the authors.

The work discussed in this paper has been funded in part by the Social Sciences and Humanities Research Council of Canada, award number 421-2015-2058 and United States Department of Homeland Security's Science and Technology Directorate under contract HSHQDC-16-C-00058. The content of the paper does not necessarily reflect the position or the policy of the Canadian or the U.S. Governments and no official endorsement should be inferred.

7. REFERENCES

[1] InterPARES. 2016. InterPARES 2 Project Glossary. URL: http://www.interpares.org/ip2/ip2_term_pdf.cfm?pdf=glossary.

[2] Geoffrey Yeo. 2007. "Concepts of Record (1): Evidence, Information, and Persistent Representations," *American Archivist* 70 No. 2 (2007), 315-43.

[3] Victoria Lemieux. 2014. "Toward a 'Third Order' Archival Interface: Research Notes on Some Theoretical and Practical Implications of Visual Explorations in the Canadian Context of Financial Electronic Records," *Archivaria* 78 (Fall 2014), 53-93.

[4] Corinne Rogers. 2016. A Literature Review of Authenticity of Records in Digital Systems: From 'Machine-Readable'to Records in the Cloud. *Acervo*, 29, 2 (2016), 16.

[5] Peter Horsman, 2002. "The Last Dance of the Phoenix, or the De-Discovery of the Archival Fonds," *Archivaria* 54 (Fall 2002): 1–23

[6] Richard Pearce-Moses. 2005. *A Glossary of Archival and Records Terminology* (Society of American Archivists, 2005).

[7] URL: http://www2.archivists.org/glossary/.

[8] Luciana Duranti. 1997. The archival bond. *Archives and Museum Informatics*, 11(3-4), 213-218.

[9] Canadian General Standards Board. 2005. National standard of Canada; CAN/CGSB-72.34-2005: Electronic records as documentary evidence. Gatineau, Quebec: National Standards of Canada.

[10] Luciana Duranti, 1998. *Diplomatics: new uses for an old science*. Scarecrow Press, Lanham, Maryland.

[11] International Organization for Standardization (ISO). 2011. TC 46/SC 11. ISO 26122: 2011. Information and documentation: work process analysis for records. Geneva, Switzerland: International Organization for Standardization.

[12] Giorgio Cencetti, "Il fondamento teorico ella dottrina archivistica," *Archivi VI* (1939), pp. 7–13 reprinted in Giorgio Cencetti, *Scritti archivistici* (Roma, 1970).

[13] Luciana Duranti, Terry Eastwood, and Heather MacNeil. 2002. *Preservation of the integrity of electronic records*. Springer Science & Business Media, Berlin.

[14] Heather MacNeil. 2008. Archivalterity: rethinking original order. *Archivaria*, 66.

[15] Sue McKemmish. 1994. Are records ever actual?. In Sue McKemmish and Michael Piggott(eds). *The Records Continuum: Ian Maclean and Australian Archives First Fifty Years.* Ancora Press, Monash.

[16] Satoshi Nakamoto, 2008. Bitcoin: A peer-to-peer electronic cash system. URL: http://www.cryptovest.co.uk/resources/Bitcoin%20paper%20Original.pdf.

[17] Coin Sciences Ltd. 2014. Metadata in the Blockchain: The OP_RETURN Explosion. URL: https://www.slideshare.net/coinspark/bitcoin-2-and-opreturns-the-blockchain-as-tcpip.

[18] Manu Sporny, Dave Longley. 2016. Flex Ledger 1.0: A flexible format and protocol for decentralized ledgers on the Web, W3C Community. URL: https://w3c.github.io/flex-ledger/.

[19] Kenneth Thibodeau. 2016. Research Issues in Archival Provenance. In Victoria Lemieux (ed). *Building Trust in Information* (pp. 69-78). Springer International Publishing, Berlin.

[20] Johannes Pfeffer. (2017). "EthOn—introducing semantic Ethereum," blogpost, https://media.consensys.net/ethon-introducing-semantic-ethereum-15f1f0696986#.ttvx7c83i.

[21] See, http://dhs2016ledger.digitalbazaar.com.

[22] Jason Bukowski, Wayne Vaughan, Shawn Wilkinson. 2016. The Chainpoint protocol, http://www.chainpoint.org/.

.