Verifying the Integrity of Information along a Supply Chain using Linked Data and Smart Contracts

Christoph Braun and Tobias Käfer

Institute AIFB, Karlsruhe Institute of Technology (KIT), Germany
uvdsl@student.kit.edu, tobias.kaefer@kit.edu

Abstract. We showcase our approach to verify off-chained information using Linked Data, Smart Contracts, and RDF graph hashes stored on a Distributed Ledger. In this demo, we present our implementation and a use case from the supply chain domain.

1 Introduction

Today's supply chain networks face the challenge of delivering goods and services not only efficiently but also transparently to the customer. Transparency gains importance especially in the food sector: Society demands more transparently provided information on products and their transportation [1]. Customers expect to find verifiable information on where food comes from and how it was treated.

Distributed Ledger Technologies (DLTs) are cryptography-based append-only databases, which are maintained on distributed nodes that need to find consensus on the state of the ledger. DLTs are designed to provide a high degree of data transparency and immutability, and gained quite some attention in recent years. However, when using DLT, issues of data sovereignty and privacy arise, as all nodes in a Distributed Ledger network store a copy of the whole ledger. This makes storing data on a ledger expensive. By "off-chaining" data [3], the data itself is stored in a database outside the ledger and only hashes are stored on the ledger as a reference. Thus, data sovereignty and privacy can be established locally and the ledger is kept at a smaller size.

The distributed ledger provides a uniform way of accessing and storing data on all nodes. For accessing and storing off-chained data, a uniform standard would be desired as well. For that, Linked Data offers a light-weight standard-based way to publish data in a decentralised fashion, where access control can be easily implemented. Hence, we ask: Can we combine the verification capabilities of the distributed ledger with Linked Data management?

In this demo, we present an implementation for our paper [2], where we address this question. The approach from our paper consists of four parts, of which an implementation of 2 and 4 are the contributions of this paper:

1. We use Linked Data, i. e. RDF and HTTP to decentrally store data off-chain. We build an item's trail of ownership, its so-called "Linked Pedigree", in the form of RDF graphs, described using the OntoPedigree ontology [5].
2. We implement a link-traversal based querying approach for verifying data on a Linked Pedigree off-chain.
3. We hash the RDF graph using the approach of [4] to connect the off-chained data with the distributed ledger.
4. We implement a Smart Contract to store RDF graph hashes in a Distributed Ledger based on Ethereum. The stored hashes are then used to verify their corresponding Linked Data.

The paper is structured as follows: First, we give a short overview on related work (Section 2). Next, we present an application showcase (Section 3) of a fisher, a trucker and a supermarket forming a supply chain to deliver a fish, i. e. a supply chain item, to an end-consumer. By this example, we show how transparent and verifiable information is generated and exchanged during the life cycle of a supply chain item. Last, we conclude (Section 4).

A website presenting a walk-through of the "LD-chain client", the CLI client implementation developed by us, and further explanation can be found online[1,2].

2 Related Work

In the intersection of supply chain and distributed ledger, IBM has launched two of the most renowned initiatives: TradeLens for global freight companies and FoodTrust for agricultural goods. Both approaches are based on Hyperledger and have similar operational characteristics: They solely rely on cryptographic access right management to ensure data privacy on the ledger. All nodes that are part of the distributed ledger hold a full copy of the ledger, where all data is stored. By including document filings, supply chain events, authority approval status and more in the ledger, scalablity may become an issue. Similarly, `provenance.org`, offering track & trace and certification of consumer goods, stores all data on a distributed ledger.

In the intersection of semantic technologies and distributed ledgers, different ontologies have been published to describe a distributed ledger, e. g. GraphChain [6], BLONDiE[3], and EthOn[4]. Our approach uses parts of EthOn. Moreover, off-chaining data has so far not been the subject of papers in the Linked Data and Distributed Ledgers workshop series (LD-DL)[5].

In the intersection of semantic technologies and supply chain, the concept of a Linked Pedigree [5] has been proposed to track & trace a supply chain item. A Linked Pedigree consists of RDF graphs that describe an item's history of

[1] `http://people.aifb.kit.edu/co1683/2019/ld-chain/semantics-demo/`
[2] `https://github.com/uvdsl/LinkedData-Logistics`
[3] `https://github.com/hedugaro/Blondie`
[4] `http://ethon.consensys.net/`
[5] Browse with `http://events.linkeddata.org/ldow-lddl/` as entry point.

ownership as Linked Data. For using the thus described data, a communication protocol is also presented in the paper. Verification is not part of the Linked Pedigree approach. In this demo, we show an implementation that adds hashing and distributed ledger technology for verification to the Linked Pedigree concept.

3 Application Showcase

To showcase our approach, we extend the motivating example from our paper [2]. We present a walk-through of this three step example of a simple supply chain:

Item Creation: A fisherman creates an item, i. e. some fish.
Item Handover: The fish is handed over between supply chain partners, e. g. from the fisherman to a trucker to a local supermarket to the consumer.
Item Verification: At the store, the consumer verifies information about the fish as a decision-making support for the purchase. Verification could also be performed during each handover.

We first introduce the architectural setup of our demonstration. Subsequently, we present a technical walk-through of our example.

3.1 Architecture

In our application showcase, we use the architecture depicted in Figure 1. Each of the three businesses, i. e. the fisher, the trucker and the supermarket, has the same setup in terms of software: A Linked Data server, e. g. a LDP[6] implementation, an Ethereum blockchain node, and the LD-chain client developed by us. The Ethereum blockchain node acts as a participant in the distributed ledger to perform data storage and computation. The LD-chain client provides a command-line interface. The end-consumer only requires an instance of the LD-chain client connected to any Ethereum node in the private network and with access to the network's Linked Data.

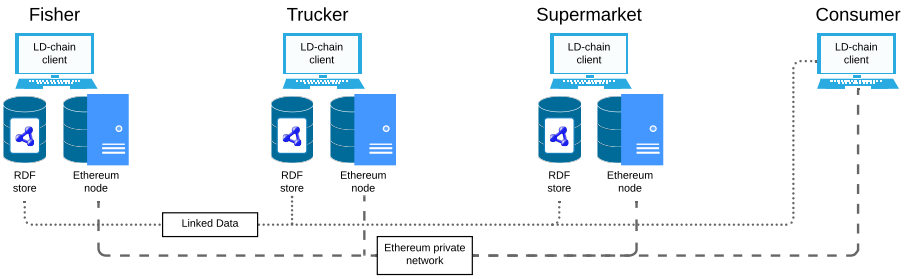

Fig. 1. The architecture we use for our demo.

[6] https://www.w3.org/TR/ldp/

3.2 Example Walk-through

The showcase depicts how transparent and verifiable information is generated and exchanged during the life cycle of a supply chain item. During the first two steps, i. e. item creation and item handover, the item's physical history is described and published as Linked Data. In the third step, i. e. item verification, we only cover the verification of the published data.

Item Creation: Initialising the Linked Pedigree. A visitor to our booth would first take the role of the fisherman and create a supply chain item by catching a fish (made of plastic). Information on the item and the catching process, e. g. fishing ground and time, is recorded in form of an RDF graph and published via HTTP. Thus, the initial part of the Linked Pedigree on the fish is formed.

Item Handover: Storing an RDF graph hash. The fisher meets up with the trucker to hand over the item. Information on the imminent item handover is included in the initial Linked Pedigree part.

Still in the role of the fisher, the visitor uses the LD-chain client to store the hash of the updated Linked Pedigree part via the Smart Contract. For that, he/she enters the URI whose content to hash. In addition, the visitor needs to specify the next owner's Ethereum wallet address. As a result, a transaction for storing the hash and transferring the digital ownership to the trucker is issued to the Ethereum node. The transaction is pending until a miner successfully mines a block that includes this transaction.

During the process, console logs are displayed about what is happening. The RDF content of the dereferenced URI can be viewed, hashing logs indicate the generated hash value, and query logs show if the submitted Linked Pedigree part is to be appended to an existing one or registered as a new initial one. When the transaction is mined, the transaction cost as well as the action performed, i. e. registering or appending, is displayed to the visitor.

Item Handover: Approving an RDF graph hash. Now, the visitor takes the role of the trucker. Before accepting the item, the visitor has to review the content of the proposed Linked Pedigree part, especially information on the imminent handover. If everything is fine, the visitor uses the LD-chain client to approve the hash of the Linked Pedigree part. For that, he/she enters the URI to approve, i. e. the URI of the fisherman's part. A transaction to approve the specified URI is issued. After successful mining, the associated cost is displayed.

Again, console logs are displayed to the visitor during the process about content, hash and verification of the input URI. When the transaction is mined, the visitor in the role of the trucker accepts to receive the physical item from the fisher.

Now the visitor can create the next Linked Pedigree part that references the initial part by its URI. Continuing the iterative example, he/she can store the new part's hash and propose the item to his/her next role, i. e. the supermarket.

Item Verification: Checking a Linked Pedigree. Skipping forward in the iterative example: The supermarket has already approved the trucker's Linked Pedigree part. The received item is now to be purchased by the end-consumer.

Taking the familiar role of the end-consumer, the visitor wants to ascertain if the fish's information has not (maliciously) been tampered with, e. g. a retrospective adjustment to the cooling information was made. To this end, the visitor uses the LD-chain client to verify the integrity of the fish's Linked Pedigree. For that, he/she enters the URI of the last known Linked Pedigree part that is provided by the supermarket. A table shows verification information of each URI dereferenced during the backwards traversal of the Linked Pedigree. This way, the customer can check the information trail on the fish until the beginning, i. e. the catchment.

Again, logs about the verification of the iteratively dereferenced URIs are displayed. This verification can be performed analogously at any point in the supply chain by any participant, starting at different points in the traversal.

4 Summary and Conclusion

We presented an implementation to verify the integrity of information along a supply chain using Linked Data and Smart Contracts. Linked Data is used to store data off the chain, and a Smart Contract manages hashes on the ledger. We showed an application to verify information along an exemplary supply chain up to the end-consumer.

We see wide application possibilities in decentrally organised logistics networks where on-premise data storage and access control is desired.

Bibliography

1. Arnot, C: Transparency Is No Longer Optional: How Food Companies Can Restore Trust, (2015). `https://www.forbes.com/sites/gmoanswers/2015/11/30/transparency-no-longer-optional/` (visited on 01/30/2019). Online
2. Braun, C and Käfer, T: Verifying the Integrity of Hyperlinked Information using Linked Data and Smart Contracts. In: Proceedings of Semantics 2019 (2019). Accepted
3. Eberhardt, J and Tai, S: On or Off the Blockchain? Insights on Off-Chaining Computation and Data. In: Proceedings of the 6th European Conference on Service-Oriented and Cloud Computing (ESOCC) (2017)
4. Hogan, A: Skolemising Blank Nodes while Preserving Isomorphism. In: Proceedings of the 24th International Conference on World Wide Web (WWW) (2015)
5. Solanki, M and Brewster, C: Consuming Linked data in Supply Chains: Enabling data visibility via Linked Pedigrees. In: Proceedings of the 4th International Workshop on Consuming Linked (COLD) at the 12th International Semantic Web Conference (ISWC) (2013)
6. Sopek, M, Gradzki, P, Kosowski, W, Kuzinski, D, Trójczak, R, and Trypuz, R: GraphChain: A Distributed Database with Explicit Semantics and Chained RDF Graphs. In: Proceedings of the 3rd Workshop on Linked Data & Distributed Ledgers (LD-DL) at the Web Conference (29th WWW) (2018)