A SoLiD App to Participate in a Scalable Semantic Supply Chain Network on the Blockchain (Demo)

Christoph H.-J. Braun and Tobias Käfer

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

Abstract. To allow for inter-organisational processes in networks with low trust, Supply Chains and corresponding information are moving to the blockchain. On the blockchain, this information poses a scalability challenge. To tackle this challenge, we propose a solution that minimises the data stored on the blockchain, which we base on semantic data modelling in knowledge graphs, decentralised management of interlinked data, and a light-weight Smart Contract. In this demo, we focus on the web agent to participate in Supply Chain networks built using our approach, and our corresponding data modelling.

Video http://people.aifb.kit.edu/co1683/2020/iswc-demo-chain/#v

1 Introduction

Blockchain-based solutions make their way into business networks with limited trust, for instance into inter-organisational logistics networks: Examples include IBM and Maersk's TradeLens[1] project or Deutsche Telekom/T-Systems and Datev's Bizzbloxx[2] project. While those solutions are gaining maturity, the scalability challenges of such solutions become apparent [3]: The inherent replication of data on the blockchain mandates that only the minimum amount of information required for fulfilling the use-case should get stored on the blockchain.

Therefore, we proposed a solution to store information off-chain and verify the information behind a supply chain using Linked Data and Smart Contracts [2]. The off-chained data is described in RDF, and accessible in a RESTful manner (i. e. Linked Data). Only hashes, URIs, and links are stored on the blockchain. The hashes, URIs, and links allow for verification of the off-chain data using a link-traversal-based querying approach and the Smart Contract, which we

[1] https://techcrunch.com/2018/08/09/ibm-teams-with-maersk-on-new-blockchain-shipping-solution/

[2] https://www.datev.de/web/de/m/presse/pressemeldungen/meldungen-2019/geschwindigkeit-und-sicherheit-im-e-commerce-mit-bizzbloxx/

presented in a previous demo at the 15th International Conference on Semantic Systems (SEMANTiCS) [1].

For the presentation for the international/virtual audience of ISWC, we want to focus on the information used around our previously presented approach, i. e. the data modelling of especially the participants' profiles and their interaction. As a showcase, we present a user agent, which consumes said RESTful APIs and implements said data modelling to manage the exchange and verification of data and the interaction among the users. Our contributions of this paper are:

- An architecture based on the practices around the decentralised social network SoLiD[3], which is based on RESTful communication, data modelling in RDF, and Linked Data Notifications as ontology and protocol to describe and interact with notifications[4]. We tie the social network to the interlinked history of a product using the standardised Activity Streams vocabulary[5].
- An app that is centred around a supply chain network participant's profile URI. The app allows for participation in the supply chain network, manages the notifications, and verifies the data using the blockchain

This paper is structured as follows: We first distinguish our current work from our previous work and provide a short overview of other related research(§ 2). In order to help the understanding of the later sections, we then cover our example use-case most briefly (§ 3). We next present our architecture and the data modelling (§ 4). Last, we conclude (§ 5).

2 Related Research

The previous version presented at SEMANTiCS was a command-line interface written in Java, which did not make use of the practices around SoLiD such as Linked Data Notifications, Activity Streams, and HTML5/JavaScript-based UIs. For the overall solution, we specifically:

- defined an ontology to describe the interlinked history of a product on the blockchain based on the ontologies OntoPedigree [6] and EthOn[6]
- connect this product history to the participating users' profiles
- make thus described data available as Linked Data
- hash the product data using the RDF hashing algorithm of [4]
- store hashes, URIs, and links in a Smart Contract on the Ethereum blockchain

The term "user profile" used in the previous version refers mainly to the Ethereum address identifying the corresponding user. In the previous version, the exchange of information and interaction among users was assumed to be managed off-chain without further specification. In contrast, the presented version specifies

[3] `http://solid.mit.edu/`
[4] `http://www.w3.org/TR/ldn`
[5] `https://www.w3.org/TR/activitystreams-vocabulary/`
[6] `https://ethon.consensys.net/`

the participants' profiles and their interaction among each other using RDF making it therefore possible to visualize or automate off-chain data exchange and verification. Based around the principles of the decentralised social network SoLiD, the current version's modelling extends the previous work regarding off-chain semantic data modelling and management.

Other previous works in the intersection of blockchain and BPM have been performed by the BPM community[7] like the seminal work around Caterpillar [5], which typically contain solutions that potentially put a comparatively high load on the blockchain, both regarding computation and storage, as they put the process logic and/or storage onto the blockchain, whereas our solution uses the blockchain only for verification and linking. For data management off-chain in the context of blockchain systems, distributed hash tables (DHTs) like the Inter-Planetary File System (IPFS)[8] are often used. Our solution however builds on Linked Data, i. e. web servers and RDF. In contrast to DHTs, web servers can implement requirements such as data sovereignty and access control in a straightforward fashion. [7] also stores data off-chain with verification capabilities, but does so at the decision of a blockchain trigger and in a place at the decision of this trigger, and not by default in a place at the decision of the user. Their approach is also built on chain code generation for pre-defined process models, whereas our approach does not build on process models and our Smart Contract code does not change.

3 Example Use-case

Our example is a supply chain network that is centred around the perishable good of a fish. The fish is caught by a fisher, transported by a trucker, and sold at a corner shop. Using our solution, we aim to allow for transparency along this supply chain by documenting the corresponding hand-overs. [2] and the video accompanying this paper contain a longer form of the example.

4 System and Data Architecture

We first describe the data pods of each network participant, before we discuss the Smart Contract and the app. The overall architecture can be found in Figure 1.

Data Pods We base our architecture on the idea of SoLiD data pods[3], where the person that owns the data pods is identified using a URI, and described using the FOAF vocabulary[9]. Following the Linked Data Notifications W3C recommendation[4], these URIs have an inbox to send requests to. These inboxes follow the Linked Data Platform[10] W3C recommendation. In our

[7] `https://www.slideshare.net/IngoWeber2/blockchain-and-bpm-reflections-on-four-years-of-research-and-applications`
[8] `https://ipfs.io/`
[9] `http://xmlns.com/foaf/spec/`
[10] `http://www.w3.org/TR/ldp`

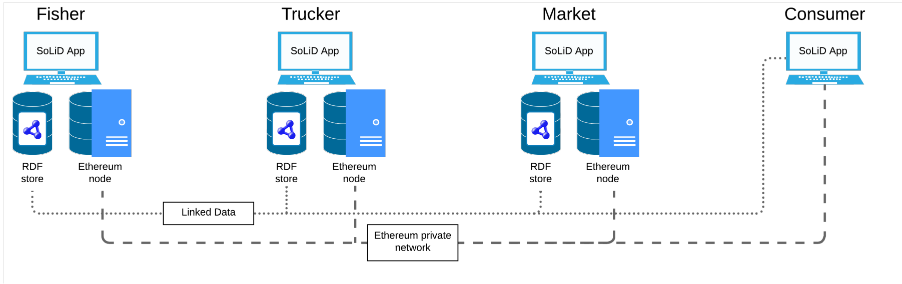

Fig. 1. The architecture of our demo. Each participant in the network has its own SoLiD pod accessed using the app. Optionally, the participants also participate with a node in the Ethereum blockchain network.

case, goods on the supply chain can be sent as offers, as defined by the Activity Stream vocabulary[5] W3C recommendation. These offers are linked to the product pedigree, and a description of how the information can be validated using the blockchain. We show a depiction of the data modelling in Figure 2.

Smart Contract Our Smart Contract for Ethereum stores (a) pairs of RDF graph hashes of product pedigree and URIs where the hashed data can be found, and (b) links to the previous pair. More details can be found in [2].

App The app takes as input the URI of the person. Then it follows hyperlinks to acquire all the required RDF data of interest for the person participating in the supply chain: The offered goods, their blockchain-based validation information, and whether the transferred good can be approved. The web agent allows a user to participate in the network and follow the hand-over and approval protocol of [2], and the verification. The web agent is built on Vue.js with RDF processing from N3.js, and interaction with Ethereum using web3.js. The agent uses the hashing algorithm of [4] in the form of an API that wraps its implementation in Java.

5 Conclusion

We presented a SoLiD app to participate in a supply chain network, where data is stored decentrally in Linked Data and verification is possible using a blockchain. The app allows for accessing the decentrally stored data using link-traversal based querying of RDF, documents the hand-overs contained in the supply chain in semantic data and on the blockchain and does the verification.

Using the app, we want to showcase the participation in blockchain-based solutions by the example of a supply chain, while addressing scalability and data sovereignty problems of blockchain technology with off-chain semantic data modelling.

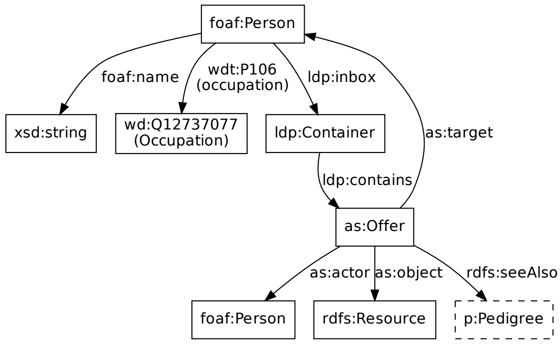

Fig. 2. The data modelling in our demo. UML classes and associations depict RDFS classes and our specialisations of RDFS domain and range respectively. The dashed part refers to the ontology that we described in detail in [2]. The URI prefixes can be looked up at `http://prefix.cc/`, besides `ethOn:`, which is short for `http://ethon.consensys.net/`, and `p:`, which is short for `http://purl.org/pedigree#`.

References

1. Braun, C.H.-J., and Käfer, T.: Verifying the Integrity of Hyperlinked Information Using Linked Data and Smart Contracts. In: Proc. 15th SEMANTiCS (2019)
2. Braun, C.H.-J., and Käfer, T.: Verifying the Integrity of Information along a Supply Chain using Linked Data and Smart Contracts. In: Proc. Posters and Demos at the 15th SEMANTiCS (2019)
3. Eberhardt, J., and Tai, S.: On or Off the Blockchain? Insights on Off-Chaining Computation and Data. In: Proc. 6th ESOCC (2017)
4. Hogan, A.: Skolemising Blank Nodes while Preserving Isomorphism. In: Proc. 24th WWW (2015)
5. López-Pintado, O., Garcıa-Bañuelos, L., Dumas, M., Weber, I., and Ponomarev, A.: Caterpillar: A business process execution engine on the Ethereum blockchain. Journal of Software Practice and Experience 49(7) (2019)
6. Solanki, M., and Brewster, C.: Consuming Linked data in Supply Chains: Enabling data visibility via Linked Pedigrees. In: Proc. 4th COLD WS @ 12th ISWC (2013)
7. Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., and Mendling, J.: Untrusted Business Process Monitoring and Execution Using Blockchain. In: Proc. 14th BPM (2016)