Using Schema.org and Solid for Linked Data-based Machine-to-Machine Sales Contract Conclusion

Xinni Wang
Karlsruhe Institute of Technology (KIT)
Karlsruhe, DE
udisg@student.kit.edu

Christoph H.-J. Braun
Karlsruhe Institute of Technology (KIT)
Karlsruhe, DE
braun@kit.edu

Andreas Both*
Technology Innovation Unit, DATEV eG
Nuremberg, DE
andreas.both@datev.de

Tobias Käfer
Karlsruhe Institute of Technology (KIT)
Karlsruhe, DE
tobias.kaefer@kit.edu

ABSTRACT

We present a demo in which two robotic arms, controlled by rule-based Linked Data agents, trade a good in Virtual Reality. The agents follow the necessary steps to conclude a sales contract under German law. To conclude the contract, the agents exchange messages between their Solid Pods. The data in the messages is modelled using suitable terms from Schema.org.

CCS CONCEPTS

• **Information systems** → **RESTful web services**; **Resource Description Framework (RDF)**; *Electronic data interchange*; • **Applied computing** → *Law*.

KEYWORDS

Solid, Agents, Linked Data, Machine-to-Machine Communication

ACM Reference Format:
Xinni Wang, Christoph H.-J. Braun, Andreas Both, and Tobias Käfer. 2022. Using Schema.org and Solid for Linked Data-based Machine-to-Machine Sales Contract Conclusion. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3487553.3524268

Video: http://people.aifb.kit.edu/co1683/2022/www-m2m/#v
Code: https://github.com/kaefer3000/m2m-solid-chain

1 INTRODUCTION

In recent and publicly announced major prototypes[1,2], Solid Pods, i. e. personal online data stores where resources are described in RDF and accessible via HTTP, serve as data stores for human agents, who can e. g. communicate by sending messages into their respective inboxes. However, it is not necessary that a Solid Pod belongs to a *human* agent. For instance, Solid Pods have recently been envisioned to serve as digital twins[3] for data aggregation from machines.

Similarly, in this demo, we want to generalise the notion of the agent associated with a Solid Pod, and have machines with Solid Pods who communicate via their inboxes. The purpose of the communication in our demo is the conclusion of a sales contract between two agents, which poses interesting questions regarding modelling and interaction between different agents and their capacities as legal personalities. However, in this demo, we focus on the implementation aspects.

Specifically, in this demo, we placed two robotic arms into a Linked Data-enabled Virtual Reality environment to trade a good. The robotic arms have writable Linked Data interfaces and the arms' behaviour is controlled via rule-based agents written in Notation3[4] syntax with ASM4LD [6] semantics, executed using the Linked Data-Fu [14] interpreter. The rule-based agents also process the data in their Solid Pods and exchange messages between their Pods, specifically the inboxes, to implement the legal steps necessary for concluding a contract to sell the good under German law. We implemented the agents to describe the payload of the messages using terms, most prominently from the schema.org[5] vocabulary. While our approach does not require the use of a blockchain for payment, we implemented the payment during the sales contract using the Ethereum blockchain.

Our contributions are thus as follows:

- The implementation of the steps between two agents to follow the steps to conclude a sales contract under German law in Notation3 rules
- The modelling of the necessary payloads using terms from schema.org
- A working demonstrator including a Virtual Reality, a User Interface to the agents' Solid inboxes, and payment

This paper is structured as follows: In Section 2, we survey related work. Next, in Section 3, we give a brief introduction of the technological foundations of our demo. Subsequently, in Section 4 we provide a walkthrough and sketch our data modelling. Hereafter, in Section 5, we describe the set-up of our demo. After that,

*Also with Leipzig University of Applied Sciences, Leipzig (Germany), andreas.both@htwk-leipzig.de
[1]https://inrupt.com/solid-enterprise-natwest-bbc
[2]https://inrupt.com/flanders-solid

[3]https://www.datasciencecentral.com/how-solid-pods-may-end-up-being-the-building-blocks-of-the-metaverse/
[4]http://www.w3.org/TeamSubmission/n3/
[5]http://schema.org/

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524268

in Section 6, we discuss our solution and outline gaps in research and technology. Lastly, in Section 7, we conclude.

2 RELATED WORK

In [1], we presented a Linked Data-enabled Virtual Reality manufacturing environment in which the production of a good is planned and executed via Linked Data. While this system was also controlled and built in a similar fashion, the setting did not contain agents with Solid Pods. Our implementation builds in part on this work.

In [2], we presented a system to offer temporary access to web resources on Solid Pods in exchange for payments via the Ethereum blockchain. While the offers and payments were announced in the Solid Pods, most of the sales process was implicitly implemented instead of following the legal steps with corresponding documents. Our implementation builds in part on this work.

How robots can act as agents to form contracts, in contrast to humans, has been the subject of scholarly investigation in robotics. For example, a discussion of the legal personality of robots, a prerequisite for the formation of contracts, was conducted by Saripan et al. in [12] who examined the attitudes of various governments in Japan, the United States of America, and Malaysia, e. g. regarding policies for robots, such as whether they are equal to humans or merely a tool.

3 PRELIMINARIES

We build our system on Linked Data, i. e. we use Uniform Resource Identifiers (URIs)[6] to denote things, the Hypertext Transfer Protocol (HTTP)[7] to interact with things, and the Resource Description Framework (RDF)[8] to describe things. In the examples in this paper, we use abbreviated URIs (CURIEs[9]) and the Turtle[10] RDF syntax for legibility. We follow the practices for URI abbreviation as curated by the community in `prefix.cc`[11].

3.1 HTTP Methods

In this demo, we use the following HTTP methods to interact with resources in request/response pairs:

GET Retrieve a resource's state's representation.

PUT Overwrite a resource's state's representation and change the resource's state accordingly.

POST Use the representation supplied in the request to create a new resource in the collection, if the request has been sent to a resource representing a collection.

DELETE Delete the resource.

3.2 Social Linked Data (Solid)

The Solid project[12] is an effort to use Web technologies, specifically Linked Data, to build a decentralised infrastructure for interaction between agents, which started with social applications [9]. In Solid, agents have a Personal Online Data store (Pod), i. e. a data store under their control using which agents can interact via the HTTP protocol. While the data in a Solid Pod is not exclusively RDF, the description of the Web resources in a Solid Pod is done in RDF. The interaction between agents follows the Linked Data Notifications (LDN)[13] protocol: An agent can send a message to another agent using an HTTP POST request (in)to the other agents' inbox, which is a collection resource (specifically, a Linked Data Platform[14] Basic Container) linked to the URI of the agent using the `ldp:inbox` property.

3.3 Schema.org

Schema.org[15] is an initiative by the big search engine providers Google, Yahoo, Microsoft, and Yandex to build a vocabulary to annotate Web pages. The vocabulary of schema.org covers a diverse set of application domains, including actions and commerce. Schema.org publishes its vocabulary using RDF-based standards accessible using HTTP.

3.4 Rule-based Linked Data agents

We specify the agent behaviour using Notation3 rules with ASM4LD semantics [6] for the Linked Data-Fu interpreter [14]. These semantics supports two types of rules:

- Rules that deduce new data, by asserting new triples once their condition holds on the data. Such rules can be used, e. g. to implement the semantics of light-weight ontology languages such as RDF Schema[16] (RDFS).
- Rules that, once their condition holds, cause the execution of an HTTP request. Such rules can be used to retrieve Linked Data (if the request is a GET request), or to enact change on Linked Data (if the request is a PUT, POST, or DELETE request, as in Figure 2), thus implementing agent behaviour.

The execution of rule sets with such semantics is done in a looped fashion. We refer the inclined reader to [6] for the details.

3.5 Ethereum

The last component of our demo is an Ethereum[17] blockchain, which our agents use to pay for the things bought. In Ethereum, agents have wallets, which are identified using a hex number. The currency in Ethereum is called ETH and wallets can contain whole or fractions of an ETH. Transactions in Ethereum can come with metadata including arbitrary text. Transactions can get sent to an Ethereum network of a set of computing nodes and get appended to the distributed ledger maintained by those nodes, if the nodes in the network find the necessary consensus. Then, the transaction (and the payment) is settled.

4 WALKTHROUGH AND ONTOLOGY

We now cover the "happy path" of the sales contract conclusion between the seller and the customer. Here, we omit details like the payment processing and cancellations (e. g. due to insufficient funds of the customer). The progression of the contract conclusion

[6] http://www.ietf.org/rfc/rfc3986.txt
[7] http://www.ietf.org/rfc/rfc7230.txt
[8] http://www.w3.org/TR/rdf11-concepts
[9] http://www.w3.org/TR/curie/
[10] http://www.w3.org/TR/turtle/
[11] http://prefix.cc/
[12] http://solidproject.org/
[13] http://www.w3.org/TR/ldn/
[14] http://www.w3.org/TR/ldp/
[15] http://schema.org/
[16] http://www.w3.org/TR/rdf-schema/
[17] http://ethereum.org/

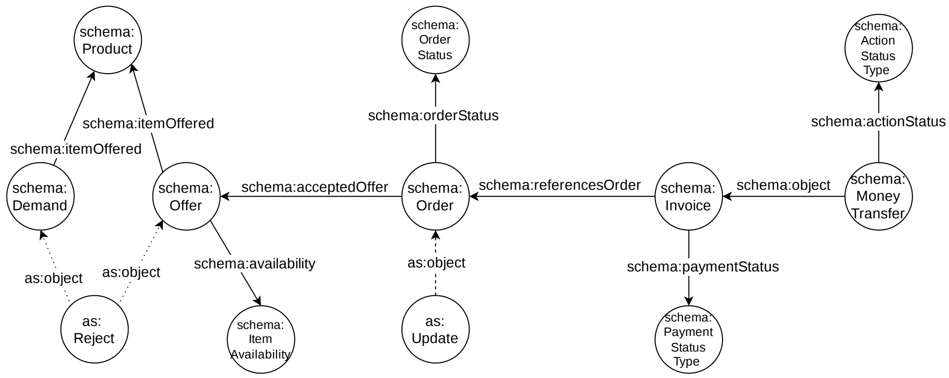

Figure 1: The involved schema.org classes during the sales contract conclusion. We depict classes as circles and use arrows to denote RDF properties that we used to connect such classes, which respect the `schema:domainIncludes` and `schema:rangeIncludes` information on schema.org for the properties. We used the vocabulary (`as:` terms) of the W3C ActivityStreams Recommendation outside of the happy path (dotted) and for technical details (dashed), where no suitable terms in schema.org were available.

forms an interlinked set of Linked Data resources. We sketch the involved classes in Figure 1.

- The customer starts with an invitation for tender for the good to be sold (lat. *invitatio ad offerendum*). The customer sends a message into the inbox of the seller using an HTTP POST request with the payload described using `schema:Demand`, which was the closest we found in schema.org. The customer links to the demanded object, which is described as being owned by the seller.
- In return, the seller sends a message to the inbox of the customer with the payload making use of the `schema:Offer` class, where s/he also notes the price of the good (see Figure 2).
- The customer accepts the offer by sending an `schema:Order` into the seller's inbox. The order links to the offer using the `schema:acceptedOffer` property.
- The seller sends a `schema:Invoice` into the inbox of the customer, linking back to the `schema:Order` in the inbox of the seller using the `schema:referencesOrder` property.
- The customer performs a payment in advance and notifies the seller using a `schema:MoneyTransfer` sent into his/her inbox. This `schema:MoneyTransfer` links to the `schema:Invoice` using the `schema:object` property. As in our demo, the payment is performed via the blockchain, the hex codes of the involved wallets are part of this message.
- The seller hands over the good and transfers the ownership, notifying the customer by updating the `schema:Order` to the status `schema:OrderPickupAvailable`.

Next to those requests, the agent controls the robot arms in the Virtual Reality to act accordingly.

We recommend the inclined reader to have a look at the video linked on the first page, which shows the progression of the data in the inboxes of both parties next to the movements of the robots and the balances of the wallets.

```
{ ?dem a schema:Demand ; schema:itemOffered ?object .
  ?seller schema:offers ?dem ; schema:owns ?object .
  ?customer schema:seeks ?dem ; ldp:inbox ?inbox . }
=> { [] a http:Request; http:mthd httpm:POST ;
       http:requestURI ?inbox ; http:body
      { <#off> a schema:Offer ; schema:itemOffered ?object ;
          schema:availability schema:InStock ;
          schema:price 20 ; schema:priceCurrency "ETH" .
        ?seller schema:offers <#off> .
        ?customer schema:seeks <#off> . } . } .
```

Figure 2: Rule to send an offer to the specified customer's inbox folder if the seller owns a demanded object.

5 SET-UP

We built our demonstrator using the publicly available Solid Pods from `solidweb.org`. This Pod provider runs the Node Solid Server[18] in version 5.6.20. As a user interface for the inboxes in the Solid Pods, we developed the SolidInboxWatcher[19], a Solid App that monitors an inbox for new and deleted entries and the entries' updates. We used Linked Data-Fu[20] 0.9.12 as agent runtime, patched for the (valid) peculiarities of the Turtle data emitted by the Node Solid Server. The Virtual Reality uses the JMonkey engine[21] in version 3.2.4-stable, for which we wrote a REST interface to the scene graph [8] executed using Jetty 9.2.29 on Java 8. We adapted the contents of the Virtual Reality for the purposes of this demo. The payments went over a privately hosted Ethereum chain, interfaced using Ganache[22] and visualised using MetaMask[23].

6 DISCUSSION

In this section, we discuss (1) the missing theoretical underpinnings and clarifications in the standards and practices that we used for this demo, and (2), how we believe that our work can serve as a foundation for an ecosystem of applications in industry.

[18]http://github.com/solid/node-solid-server
[19]http://github.com/uvdsl/solid-inbox-watcher
[20]http://linked-data-fu.github.io/
[21]http://jmonkeyengine.org/
[22]http://trufflesuite.com/ganache/
[23]http://metamask.io/

6.1 Missing foundations

The following discussion boils down to the question: What does it mean to send a POST request into a Solid inbox?

6.1.1 Data modelling. If we interpret the inbox of an agent as a mailbox, we are inclined to use the terms from our physical lives for the documents that we send via mail for what we send using POST requests into inboxes. In our demo, think e. g. of the `schema:Order`, and look at the right column in Table 1. That would be in line with the function of the POST request "Posting a message" [4].

If, however, we consider the agent as a process that does things, we rather would like to send *actions* in a POST request that we think the owner of the Solid Pod should do. For example, in our demo, think of the act of ordering a good, for which we could use `schema:OrderAction` from the left column of Table 1. This action would of course have to be sent in the other direction as the `schema:Order` in our demo, and a gatekeeper would be required on the inbox such that only such actions can get successfully posted to the inbox which the agent would be OK with committing themselves to performing. That would be in line with the function of the POST request "Providing a block of data, such as the fields entered into an HTML form, to a data-handling process" [4].

6.1.2 Philosophy of language and the law. Semantic underpinnings are also missing when it comes to the expectations that a recipient of e. g. a confirmed `schema:Order` can have. Has the agent who confirmed the order legally committed themselves to provide a good or service? In other words, has the agent performed a commissive speech act [13]? That may require to specify semantics for data exchange, as shown in this demo, similar to what the multi-agent systems community has done around the FIPA Agent Communications Language [11], which already had been discussed for Web services [5]. On top, has the agent the legal capacity to form a contract? Such questions are also discussed in robotics [3, 12].

6.2 Application potential

From an industry perspective, the main potential of work such as ours, is to enable digital ecosystems across domains, countries, and organisations. A core limitation of current industrial scalability is the effort required to align data formats and service interfaces for interoperability. We believe that the adherence to the Linked Data principles reduces such effort: Given the extensible data format of RDF, enterprises can, at the same time, define use-case specific data models while adhering to globally accepted terminology. Given the ubiquitous HTTP protocol, which builds on the internet, enterprises can interact with software around the globe. As our demo shows, vocabularies with well-defined semantics are a cornerstone for real-world use, e. g. when processes that stem from the legal profession are to be implemented. Besides schema.org, vocabularies like those defined in [15] and [10] may serve as foundations of a growing Linked Data ecosystem. Next to vocabularies, we believe that a cross-organisation ecosystem requires standardised sequences of interactions, as shown in this demo, such that business models of different organisations can get integrated using Linked Data. Lightweight process models such as [7] can be the next step. Solid paves the way, e. g. by building the necessary access control mechanisms.

Action	Result
`schema:InviteAction`, `AskAction`	`schema:Demand`
`schema:QuoteAction`	`schema:Offer`
`schema:OrderAction`, `AcceptAction`	`schema:Order`
`schema:ConfirmAction`	`schema:Invoice`

Table 1: A rough correspondence between terms for actions and their results in the schema.org vocabulary.

7 CONCLUSION

We presented a demo in which agents conclude a sales contract by communicating via Solid Pods. We described the technical foundations insofar as they are specified and provided a walk-through for our demo next to our data modelling. After that, we pointed out gaps in the technical foundations and hinted at possible solutions. Lastly, we gave a perspective on how our work can be regarded as a step towards cross-organisational digital ecosystems.

ACKNOWLEDGMENTS

This work has been supported in part by the German ministry BMBF under grant 16DTM107B ("MANDAT"). We acknowledge the help of Jan Henze, from whose prototype we took inspiration.

REFERENCES

[1] Nico Aßfalg, Helen Schneider, and Tobias Käfer. 2022. Integrated Planning and Execution on Read-Write Linked Data. In *Proceedings of the All the Agents Challgenge (ATAC) at the 20th International Semantic Web Conference (ISWC 2021).*

[2] Hendrik Becker, Hung Vu, Anett Katzenbach, Christoph H-J Braun, and Tobias Käfer. 2021. Monetising Resources on a Solid Pod using Blockchain Transactions. In *Proceedings of Demos at the 18th Extended Semantic Web Conference (ESWC).*

[3] Peggy Doerschuk, Ling Wang, and Amit Keswani. 2006. Communication Language for Autonomous Multi-Robot Systems. In *Proceedings of the International Conference of Robotics and Applications (IASTED-RA 2006). Actapress.*

[4] Roy Fielding and Julian Reschke. 2014. Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. IETF RFC7231.

[5] Nicholas Gibbins, Stephen Harris, and Nigel Shadbolt. 2003. Agent-Based Semantic Web Services. In *Proceedings of the 12th International Conference on World Wide Web (WWW).* 710–717.

[6] Tobias Käfer and Andreas Harth. 2018. Rule-based programming of user agents for linked data. In *Proceedings of the 11th Workshop on Linked Data on the Web (LDOW) at the 27th Web Conference (WWW).*

[7] Tobias Käfer, Benjamin Jochum, Nico Aßfalg, and Leonard Nürnberg. 2021. Specifying and executing user agents in an environment of reasoning and RESTful systems using the guard-stage-milestone approach. *J. Data Semant.* 10, 1-2 (2021).

[8] Felix Leif Keppmann, Tobias Käfer, Steffen Stadtmüller, René Schubotz, and Andreas Harth. 2014. High Performance Linked Data Processing for Virtual Reality Environments. In *Proceedings of the Posters & Demonstrations Track at the 13th International Semantic Web Conference, (ISWC).*

[9] Essam Mansour, Andrei Vlad Sambra, Sandro Hawke, Maged Zereba, Sarven Capadisli, Abdurrahman Ghanem, Ashraf Aboulnaga, and Tim Berners-Lee. 2016. A Demonstration of the Solid Platform for Social Web Applications. In *Proceedings of Posters & Demos at the 25th Int'l Conference on World Wide Web (WWW).*

[10] Boris Otto, Michael ten Hompel, and Stefan Wrobel. 2019. International data spaces. In *Digital Transformation.* Springer, 109–128.

[11] Stefan Poslad. 2007. Specifying Protocols for Multi-Agent Systems Interaction. *ACM Trans. Auton. Adapt. Syst.* 2, 4 (nov 2007), 15–es.

[12] Hartini Saripan, Nurus Sakinatul Fikriah Mohd Shith Putera, and Nurus Sakinatul Fikriah. 2016. Are robots human? A review of the legal personality model. *World Applied Sciences Journal* 34, 6 (2016), 824–831.

[13] John Rogers Searle. 1969. *Speech acts: An essay in the philosophy of language.* Vol. 626. Cambridge University Press.

[14] Steffen Stadtmüller, Sebastian Speiser, Andreas Harth, and Rudi Studer. 2013. Data-Fu: a language and an interpreter for interaction with read/write Linked Data. In *Proceedings of the 22nd International World Wide Web Conference (WWW).*

[15] Wouter Termont, Wouter Janssens, Ruben Verborgh, and Tom Haegemans. 2021. A Decentralised Registry to Support the Dynamic Binding of Semantic Components in Solid Data Browsers. *Digita Research Report DGT_2* (2021), 1–1.