Towards Semantics and Protocols for Contract Conclusion via the Web Architecture – A Gap Analysis

Xinni Wang[1], Tobias Käfer[1,2]

[1]Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
[2]Alexander von Humboldt Institute for Internet and Society (HIIG), Berlin, Germany

Abstract
We investigate how contracts can get concluded via the Web architecture. While other fields of research such as multi-agent systems or e-commerce have investigated different aspects of the problem, the peculiarities of the Web architectures pose specific challenges. We identify different aspects of the problem (interaction, message content, and codification), discuss how different aspects of the problem are tackled in protocols by different communities, and describe gaps for solutions based on Semantic Web technologies. We close with a call to action for corresponding legislation/jurisprudence, ontology and interaction protocol development to build in the necessary semantics.

This paper integrates with the W3C Dataspaces CG by addressing the generic dataspaces issue No. 9.

1. Introduction

Business transactions are moving online and getting increasingly automated: Already before the COVID-19 pandemic, Forrester, a market research company, has estimated the market for electronically concluded B2B transactions as being at the size of 18 trn USD by 2021[1]. On top, automatic conclusion of transaction becomes increasingly relevant if we consider, e. g. recent standardisation around administration shells for Industry 4.0 component interaction [2], the need for data sharing agreements to fulfil the requirements of the GDPR [3], ongoing standardisation around Data Spaces[2], or the growing market around charging electric vehicles: Precedence research, another market research company, expects the market for the latter to be at 7 bn USD and growing at 28 % p. a.[3].

However, a prerequisite for the automated conclusion of transactions is a common understanding of how transactions come about, i. e. a protocol that consists in messages that are exchanged in interaction steps, with clear semantics and codified ramifications. In this paper, we therefore wonder: *do Web technologies provide the necessary building blocks in semantics of both data and interaction to facilitate contracting?* Issues around how actors are identified, contracts are signed and archived, are also important for contracting, but out of the scope of this paper.

The knowledge graph- and web-based substrate for such transactions is building up: already today, solutions based on Knowledge Graphs and the Web are under development for asset administration shells [1], GDPR[4] and Data Spaces[2]. On top, we observe that the substrate for Web-based transactions between agents is materialising: Solid's WebIDs[5] with inboxes (the latter based on the W3C Recommendation Linked Data Notifications [7]) have been identified as a way to implement embodied agents on the Web [41], and research is being done to make this substrate ready for data exchange in business-to-business settings [4].

Previously, such semantics and protocols have been encoded in framework contracts, e. g. UN/EDIFACT interchange agreements [37], under which then electronic messages with the corresponding semantics

The Third International Workshop on Semantics in Dataspaces, co-located with the Extended Semantic Web Conference, June 01, 2025, Portorož, Slovenia.

[1]https://www.forrester.com/report/RES136173
[2]https://docs.internationaldataspaces.org/ids-knowledgebase/dataspace-protocol
[3]https://www.precedenceresearch.com/electric-vehicle-charger-market
[4]https://w3c.github.io/dpv/dpv/
[5]https://solid.github.io/webid-profile/

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

are exchanged in a protocol. Due to their cost and effort to set-up, we assume that such framework contracts are most relevant in cases where big companies with a high volume of transactions interact. However, we believe that with the growing number of transactions online that also involve smaller parties who communicate increasingly using decentralised technologies, such framework agreements are less feasible and we need to put more semantics into the infrastructure.

Semantics and protocols for the electronic conclusion of contracts by communicating via the Web touches a number of disciplines and standards, some of which are recent developments. We base our discussion on works from the multi-agent community, e. g. [21, 25], which recently tightened the connection to Web research [8]. We discuss established standards such as UN/EDIFACT [9] and FIPA [13], next to recent developments in Industry 4.0 protocols [2], and on the Web such as ActivityPub [39] and Solid[6], which is currently being standardised by the Linked Web Storage Working Group at the W3C. Moreover, the recent Open Digital Rights Language [23] allows to model obligations, and its semantics are still under development [19]. We note that our work echos ideas from the Pragmatic Web vision [30, 32]. We do not discuss so-called Smart Contracts, which are merely code that runs on a distributed ledger, and their fit to contracts can be questioned from a legal perspective [20].

We assume a basic familiarity of the reader with basics from the legal and the web realm: First, with the legal concept of a contract, which is being formed by two agents interacting by exchanging so-called *declarations of intentions* in messages. Hereby, it is not just important that the *message content* contains the right information (e. g. so-called essentialia negotii) – hinting at semantics of data, but also that the *interaction* is done with the right intentions, in the right order, and under the right circumstances (e. g. requirements of form) – hinting at the semantics of interaction in a protocol. The steps to form a contract can be: invitation to treat, offer, acceptance, payment, handover, and transfer of ownership. Second, we assume familiarity with URIs[7], CURIEs[8] abbreviated according to `prefix.cc`, and HTTP [12].

The paper is structured as follows: In Section 2, we present different digital works on and off the web that addressed the conclusion of contracts or at least can provide useful building blocks for re-building contract conclusion on the Web. We next compare (Section 3) and discuss those works, and place a special focus on Web standards (Section 4). In Section 5, we conclude with a call to action.

2. Standards and Approaches for Electronic Transactions

We survey different standards and approaches for electronic agent-to-agent transactions. We look at how agents communicate, especially *interaction and message* content, next to the foundations in terms of the law (i. e. *codification*) and/or philosophy of language, specifically Searle's Speech Act theory [31], a theory for how humans act by the means of different types of utterances.

2.1. Standards of the Foundation for Intelligent Physical Agents (FIPA)

The Foundation for Intelligent Physical Agents (FIPA) is an organisation to promote agent-based technologies. In the late 1990s and early 2000s, they built standards for inter-agent communication. Those standards include the FIPA Message-transport Service (MTS) for the message transport between inter-operating agents [16], the FIPA Agent Communications Language (ACL) for how to express message parameters [13], next to a grounding of ACL in Speech Acts, the FIPA communicative acts [17]. Next to those basic standards, FIPA includes a protocol for contract conclusion using their set of standards. FIPA provides the following building blocks in our three areas:

Interaction The FIPA standards list the following ways for communication between agents. They may or may not use an Agent Communication Channel (ACC), and abstract away different Message Transport Protocol (MTP) implementation options.

- **Via multiple ACCs** Agent A → local ACC → remote ACC using suitable MTP → Agent B

[6] `https://solidproject.org/`
[7] `https://www.ietf.org/rfc/rfc3986.txt.`
[8] `https://www.w3.org/TR/curie/`

- **Via one ACC** Agent A → remote ACC using suitable MTP → Agent B
- **Direct communication** Agent A → Agent B (only listed, but not covered in the standard)

The standard does not specify the final message delivery by the ACC.

Implementation options for the MTP include CORBA's Internet Inter-ORB Protocol (IIOP) [15] or tunnelling messages through HTTP POST requests [14] targeted at the ACC.

Message Content To communicate, agents transfer ACL messages, which can consist of different parameters incl. performative, sender, receiver, content, language, protocol. While only the performative parameter is mandatory for an ACL message, most messages also contain sender, receiver and content parameters. The performative parameter specifies the FIPA communication act for the message.

Using the **FIPA Contract Net Interaction Protocol (CN)** [18], contracts can be concluded. The protocol can be used in the protocol parameter of an ACL message by the token fipa-contract-net. It starts with an agent, the *initiator*, issuing a call for proposal (cfp) other agents, the so-called *participants*. Each participant can either refuse the cfp or propose with the preconditions that the participant is setting out for the task (e. g. time, price). Once a given deadline passed, the initiator evaluates the received proposals. For each proposal, the initiator accepts/rejects the proposal (accept-proposal/reject-proposal communication act). The proposals are binding, so an proposal acceptance by the initiator leads to the acquisition of the participant's commitment to perform. Once the participant completes, it sends a completion (inform-done or inform-result) or a failure message. This protocol is basis of an Industry 4.0 standard [2], but does not have a legal grounding. Speech Act-based semantics can drawn from [17].

2.2. ActivityPub Protocol

The ActivityPub Protocol [39] is a W3C Recommendation and provides a decentralized social networking protocol based on the Activity Vocabulary [33]. ActivityPub is, e. g. the basis for communication in the federated social networking software Mastodon[9].

Users in ActivityPub are represented by "actors". Actors have an inbox as well as an outbox, both identified using URIs, on a Web server. Both the inbox and the outbox are defined as `as:OrderedCollection`, which contain all the messages received (resp. produced) by the actor. HTTP POST and GET requests can be used to append to or read messages from inbox and outbox.

The ActivityPub protocol offers the following building blocks in our areas:

Interaction ActivityPub has two interaction layers: Social API and Federation Protocol:

The **Social API** is a client-to-server protocol that permits a client to act on behalf of an user. A client communicates with a server by posting to the user's outbox on the server via HTTP POST requests.

The **Federation Protocol** is a server-to-server protocol and is used to distribute messages between actors. The server to server communication is typically triggered by a client-to-server interaction and facilitates the delivery of the content of the outboxes. A server communicates with another server by posting to an actor's inbox on the other server via HTTP POST requests.

Message Content The ActivityPub protocol is about interaction in a social network (e. g. liking messages, following and blocking other users) and thus uses terms from the Activity Vocabulary [33]. Therefore, the contents of inboxes and outboxes are instances of `as:Activity`. If a user puts an object into their outbox that is not an activity (e. g. a blog post), then the server will wrap the object in a `as:Create` activity, thus recording the creation of the object. Different subtypes of activity include `as:Invite` or `as:Reject`. The Actitvity Vocabulary does not come with semantics and from the tenses used in the notes from the documentation and the examples is unclear to us, e. g., whether instances of the vocabulary's classes refer to the past (like in a log), or to the present (a description of the state of affairs or a command).

[9] https://joinmastodon.org/

2.3. United Nations rules for Electronic Data Interchange For Administration, Commerce and Transport (UN/EDIFACT)

UN/EDIFACT is an international standard for electronic data interchange (EDI) by the United Nations Economic Commission for Europe (UNECE). It comprises a set of internationally agreed standards, directories and guidelines for the electronic interchange of structured data. The interchange particularly relates to trade in goods and services between independent, computerized information systems [36]. UNECE provides a multitude of standards and specifications for relevant processes in the supply chain management domain such as the quotation process [10], the ordering process [9] or the despatch and receive process [11]. In the areas we identified, UN/EDIFACT provides the following building blocks:

Interaction UN/EDIFACT messages can be transferred via many protocols, including FTP, SMTP, and HTTP. HTTP is being used with clarifications about its application (e. g. regarding encryption) in Applicability Statement 2 (AS2) [28]. AS2 uses the HTTP POST request to tunnel data. The standards demand trading parties to create agreements that specify the delivery of messages (FTP server and directory, target URI of HTTP POST requests).

Message Content There are message types for many business-relevant messages, including request for quote, order, invoice, despatch avis, among others, updated until the present day. UN/EDIFACT messages are written in a text-based and hierarchically structured format, by default in 7-bit ISO 646. There is also an XML-based version, XML/EDIFACT.

Codification The UN/EDIFACT standards covers the question what is required for the EDI messages to have a legally binding effect. They solve the codification problem by stating that the trading parties need an interchange agreement. These interchange agreeements cover issues such as [37]:

- The selection of EDI messages, message standards and methods of communication (e. g. via FTP, HTTP or AS2)
- The points at which EDI messages have legal effect
- The laws governing the interchange of EDI messages.

2.4. Legal Agent Communication Architecture (LACA)

The authors of [21] studied the Dutch General Act on Administrative Law (Algemene Wet Bestuursrecht, AWB) law from an agent perspective. They proposed the Legal Agent Communication Architecture (LACA) using which they model agents that communicate. They derive a number of communication primitives from the AWB and provide an underpinning with Searle's speech acts.

The discussion in [21] is more on the conceptual level, thus in terms of building blocks for our areas, we can only report abstract concepts without implementation:

Interaction Each agent has a working memory into which other agents or the human on whose behalf the agent acts can post communication primitives.

Message Content Messages contain one or more communication primitives, the address of the recipient and the name of the sender.

Communication primitives in LACA use four of the five speech act of Searle: assertive, commisive, declarative and directive communication primitives, which implement the basic procedures of the AWB.

2.5. International Data Spaces (IDS)

There is ongoing work on the Dataspace Protocol [26] at the Eclipse foundation driven by the International Data Spaces initiative. While the work is preliminary, we discuss it here briefly, as the protocol is dubbed *Release Candidate* such that we assume reasonable maturity, and due to the popularity of International Data Spaces. The Dataspace Protocol aims at forming data sharing agreements.

Interaction Parties in an agreement to be formed exchange messages to evolve a state machine for agreement forming. Thereby, parties are identified using URNs or UUIDs. There is a binding of the messages exchanges for state machine evolution to an HTTP-based interaction with endpoints for the different states. Thereby, messages are tunnelled via HTTP, prominently using the POST request.

Message Content Messages are described in JSON-LD, governed by JSON Schemas. The messages wrap policies such as offers and agreements. While those wrapped policies use words from ODRL, the semantics are unclear, as ODRL is not normative for the protocol. While JSON schemas for the messages exist, it is unclear whether they cover all essentialia negotii for valid data sharing agreements [6].

2.6. M2M Trading using Solid (M2M-Solid)

In previous work, we [38] presented a demonstrator in which we implemented a Linked Data-based machine-to-machine sales contract conclusion that follows the necessary legal steps under German law. We made the following design decision in the identified areas:

Interaction In Solid, agents are identified using WebIDs (i. e. URIs) and communicate by exchanging messages between inboxes on their respective Solid Pod. A Pod is a data store under an agent's control, which can get accessed RESTfully using the HTTP protocol. The inboxes follow the Linked Data Notifications (LDN) protocol [7]. Solid has originally been developed for social interactions [27].

Message Content The content of the messages exchanged between the inboxes is modelled predominantly using the Schema.org vocabulary. Schema.org [5] is an initiative by the 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 domains, including actions and electronic commerce. The part of schema.org electronic commerce is based on the GoodRelations vocabulary [22]. Schema.org covers, e. g. offers, orders, and invoices, and leaves open whether they are future- or backward-looking. Where Schema.org did not provide the right or closely related terms, we resorted to the Activity Vocabulary.

3. Comparison of FIPA ACL, ActivityPub, UN/EDIFACT, LACA, IDS and M2M-Solid

We next compare the presented approaches regarding the categories interaction, message content, and codification, with a special focus on fit to the Web.

Interaction In Table 1, we compare FIPA ACL, ActivityPub, UN/EDIFACT, LACA, IDS, and M2M-Solid regarding the message exchange architecture. All approaches except LACA come with an implementation or specification of how messages are exchanged. LACA merely mentions a "working memory".

ActivityPub and M2M-Solid have been built for the Web and make use of HTTP in a RESTful manner. While the other approaches leave the option to operate via HTTP, they typically do not make use of the Web architecture and HTTP's message semantics, but instead merely use the HTTP POST request to channel data to a message broker. Correspondingly, for ActivityPub and M2M-Solid there is a clear notion of an inbox, which allows for the agents to do asynchronous processing of messages that other agents posted into the inbox. For the approaches with a message channel such as FIPA, the delivery to agents and their processing is not specified. UN/EDIFACT supports some sort of inbox based on the communication channel agreed in the trading partners' interchange agreement: e. g. if the parties decided on folders on FTP servers, this is where they deposit requests and acknowledgements. IDS uses individual endpoints for each of the different message types. The working memory in LACA can be regarded as some sort of inbox and outbox, since the authors of [21] state that the working memory can be seen as a blackboard where LACA communication primitives can be posted onto. The LACA communication manager could be compared to the ActivityPub's Federation Service.

Message Content In Table 2, we present the legal steps in a sales contract process next to terms from the vocabularies of the different approaches. For FIPA, there is the Contract Net Interaction Protocol, which covers all the steps and has therefore matching vocabulary for each phase. Similarly, UN/EDIFACT focuses on interchanges related to the trade in goods and offers a fine-grained vocabulary

Table 1
Comparison of standards and approaches in the interaction area.

	FIPA ACL	ActivityPub	UN/EDIFACT	LACA	IDSA	M2M-Solid
Commu-nication Method	CORBA, HTTP POST	REST/ HTTP	SMTP, FTP, AS2/HTTP POST	communication manager (not specified)	endpoints (HTTP binding)	REST/ HTTP
Inbox	(ACC)	`as:OrderedCollection`	FTP server or some process	working memory	endpoint	`ldp:BasicContainer`
Outbox	(ACC)	`as:OrderedCollection`	–	working memory	–	–

[35] to match the legal steps. LACA does not have e-commerce specific terms, but more general terms from the law, thus we used close matches for each step. For the invitation to treat step, we used the term query instead of request, since the acceptance of a request, in this case by making an offer, would have the perlocution that the requesting agent becomes committed to the request. The legal and speech act-based semantics of LACA allow us to make such arguments. The semantic web vocabularies we found are typically not built to engage in legal transactions, be it around commerce or in general: For ActivityPub, we thus used the closest matches in the Activity Vocabulary, which has been built for social networks. For the invitation to treat, we chose `as:Announce` which "indicates that the actor is calling the target's attention the object." [33]. This definition does not fit the invitation to treat step since it lacks the aspect of the directive speech act of asking the addressee to make an offer, yet it was the closest match we found. Other steps such as payment and handover are matched with the same closest-matching higher-level term. While ActivityPub is based upon the Activity Vocabulary, ActivityPub only uses parts of the Activity Vocabulary. Some terms that would have been useful for our purposes, e. g. `as:Offer` are not defined in ActivityPub. On top, `as:Announce` is only defined in the Federation Protocol but not the Social API, i. e. not for the outgoing communication of an agent or user. IDS wraps ODRL-inspired offers and agreements in messages that –with the right semantics– are useful for contracting. M2M-Solid is based upon the schema.org vocabulary, which contains suitable e-commerce terms with textual definitions that fit our purposes. Where schema.org does not have appropriate terms, M2M-Solid uses terms from the Activity Vocabulary.

Codification UN/EDIFACT describes how legally binding agreements can be formed. LACA is an interpretation of the Dutch law.

4. Semantic Web Technologies, Discussed

We discuss the presented approaches for their transferability to the Web.

Interaction ActivityPub and Linked Data Notifications support inboxes using which agents can write using HTTP POST. That would be in line with the function of the POST request "Posting a message" [12]. In Solid, agents are identified using WebIDs [29] and advertise their inbox according to LDN using the `ldp:inbox` property. However, the WebID specification has not reached W3C Recommendation status yet. In the Decentralised Identifier (DID) W3C Recommendation [34], DID subjects can advertise services for interaction. However, those services are optional, thus we cannot uniformly assume that we can deliver messages to DID-identified agents.

Message Content Different vocabularies use subtly different semantics for their terms regarding the temporal aspects of what they describe: Backward-looking vocabularies such as the SOSA ontology [24] define their terms explicitly as for logging a *past* event: For instance, a `sosa:Actuation` describes an actuation that has happened in the past. Forward-looking terms such as UN/EDIFACT's advice messages talk about events that are going to happen in the *future*: The despatch advice message (DESADV)

Table 2
Mapping of vocabulary between different standards and approaches in the message content area. Handovers refer to the good and the ownership. Neg – Negotiation, Term – Termination, Ev – Event, Msg – Message.

Legal process	FIPA ACL	Activity Pub	UN/EDIFACT	LACA	IDS	M2M-Solid
Invitation to treat	cfp	`as:Announce`	request for quotation (REQOTE)	query	`ds:ContractRequestMsg`	`schema:Demand`
Offer	propose	`as:Offer`	quote (QUOTES)	request	`ds:ContractOfferMsg`	`schema:Offer`
No offer	refuse	`as:Reject`	no response	reject	`ds:ContractNegTermMsg`	`as:Reject`
Acceptance	accept-proposal	`as:Accept`	purchase order (ORDERS)	accept	`ds:ContractAgreementMsg`	`schema:Order`
Rejection	reject-proposal	`as:Reject`	no response	reject	`ds:ContractNegTermMsg`	`as:Reject`
Payment	inform	`as:Announce`	remittance advice (REMADV)	inform	–	`schema:MoneyTransfer`
Handovers	inform	`as:Announce`	despatch advice (DESADV)	inform	`ds:ContractNegEvMsg`	`as:Update`
Payment error	failure	`as:Update`	application error (APERAK)	cancel	–	`as:Update`

Table 3
Legal steps, different modelling approaches and corresponding speech acts. Handovers refer to both the good and the ownership. SA – Speech Acts.

Legal process	M2M-Solid	M2M-Solid SA	LACA	LACA SA
Invitation to treat	`schema:Demand`	directive	query	directive
Offer	`schema:Offer`	directive+commissive	request	directive
No offer	`as:Reject`	expressive	reject	commissive
Acceptance	`schema:Order`	directive+commissive	accept	commissive
Rejection	`as:Reject`	expressive	reject	commissive
Payment	`schema:MoneyTransfer`	assertive	inform	assertive
Handovers	`as:Update`	assertive	inform	assertive
Payment error	`as:Update`	assertive	cancel	assertive

models a future despatch. The schema.org vocabulary is somewhere in between: `schema:Action` can have an `schema:ActionStatus` attached that indicates whether, e. g., the action could get triggered or has already completed. The Activity Vocabulary's documentation is contradicting: While the examples have explanations that indicate that the terms are about the past, the explanations of the terms are formulated in different tenses. IDSA uses ODRL-inspired terms and thus does not formally build on the meaning of ODRL terms. A normative relation to ODRL would help. None of the surveyed Semantic Web vocabularies was sufficient for our purposes. Schema.org only contains terms for parts of the process, and the Activity Vocabulary's terms were not precise enough.

Codification We have seen different approaches that come with different semantic underpinnings both in the law and speech acts: FIPA presented a rich vocabulary and specified the underlying semantics based on Speech Act theory, about which there is academic debate, e. g. [40, 25]. UN/EDIFACT addresses the need for legal underpinnings of the data exchange. For legal implications to take effect, interchange agreements between the trading partners have to be negotiated first, which need to define the implications of message sending and reception. LACA started out with the law and defined the semantics of messages based on Speech Acts. Table 3 presents the legal process, a speech act interpretation for M2M-Solid, next to the closest match for the legal steps we found in LACA's speech acts.

5. Conclusion and Call to Action

Using this paper, we set out to look at the state of the art in modelling and protocols for contract conclusion via the web. We summarise our findings in three areas in a call to fill the gaps we identified:

Interaction To engage in a contracting protocol, we need to make valid declarations of intention via the Web, thus Web standards need to state that we can send messages with (legal/speech act) semantics, to inboxes reachable over HTTP. Inspiration for such semantics can be drawn from LACA, FIPA, and the interchange agreements from UN/EDIFACT.

Message Content On the level of vocabularies and ontologies, the definitions of terms need to get strengthened in order to know whether the terms are suitable to engage in protocols for legal transactions. We have met schema.org as a good, but for our purposes incomplete example, and ActivityVocabulary as an ambiguous example. We also discussed non-(Semantic-)Web approaches such as the FIPA CN and UN/EDIFACT that provide terms with useful semantics.

Codification In the legal domain, the rules for the digital realm should get updated and internationally coordinated: To deliver declarations of intention to recipients, the legally uniformly viable standards are still mail and fax, thus the law should open up digital options. For instance, in Germany, since 2023 some government organisations [§173 Zivilprozessordnung] and since 2024 also some business entities are required to offer a secure electronic delivery channel.

Declaration on Generative Artificial Intelligence

The authors did not use generative artificial intelligence in the writing of this paper.

References

[1] S. R. Bader et al. *What is the Asset Administration Shell from a technical perspective?* Tech. rep. Plattform Industrie 4.0, 2021.

[2] O. Bieliaiev et al. *Language for I4.0 components – Interaction protocol for tendering procedures.* Richtlinie VDI/VDE 2193. Verein Deutscher Ingenieure (VDI), 2019.

[3] A. Both, T. Kastner, D. Yeboah, C. Braun, D. Schraudner, S. Schmid, T. Käfer, and A. Harth. "AuthApp - Portable, Reusable Solid App for GDPR-Compliant Access Granting". In: *Proceedings of the 24th International Conference on Web Engineering (ICWE).* 2024.

[4] A. Both, T. Kastner, D. Yeboah, C. H.-J. Braun, D. Schraudner, S. Schmid, T. Käfer, and A. Harth. "Foundational Components for B2B Data Sharing using the Solid Protocol". In: *Journal of Web Engineering* (2025). Accepted.

[5] D. Brickley, ed. *schema.org*. URL: https://schema.org/.

[6] C. Caimi, C. Gambardella, M. Manea, M. Petrocchi, and D. Stella. "Legal and Technical Perspectives in Data Sharing Agreements Definition". In: *Proc. 3rd Annual Privacy Forum (APF).* 2015.

[7] S. Capadisli and A. Guy. *Linked Data Notifications.* Recommendation. W3C, May 2017. URL: https://www.w3.org/TR/ldn/.

[8] A. Ciortea, S. Mayer, F. Gandon, O. Boissier, A. Ricci, and A. Zimmermann. "A Decade in Hindsight: The Missing Bridge Between Multi-Agent Systems and the World Wide Web". In: *Proceedings of the 18th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS).* 2019.

[9] *Cross Industry Ordering Process.* Business Requirements Specification. UN/CEFACT, 2008.

[10] *Cross Industry Quotation Process.* Business Requirements Specification. UN/CEFACT, 2010.

[11] *Cross-Industry Despatch and Receive Process.* Business Requirements Specification. UN/CEFACT, 2021.

[12] R. Fielding and J. Reschke. *Hypertext transfer protocol (HTTP/1.1): Semantics and content.* RFC 7231. IETF, June 2014. URL: https://www.ietf.org/rfc/rfc7231.txt.

[13] *FIPA ACL Message Structure Specification.* Standard SC00061G. FIPA, 2002.

[14] *FIPA Agent Message Transport Protocol for HTTP Specification.* Standard SC00084F. FIPA, 2002.

[15] *FIPA Agent Message Transport Protocol for IIOP Specification.* Standard SC00075G. FIPA, 2002.

[16] *FIPA Agent Message Transport Service Specification.* Standard SC00067F. FIPA, 2002.

[17] *FIPA Communicative Act Library Specification.* Standard SC00037J. FIPA, 2002.

[18] *FIPA Contract Net Interaction Protocol Specification.* Standard SC00029H. FIPA, 2002.

[19] N. Fornara, V. Rodríguez-Doncel, and B. Esteves. *ODRL Formal Semantics.* Draft Community Group Report. ODRL CG, 2025. URL: https://w3c.github.io/odrl/formal-semantics/.

[20] M. Giancaspro. "Is a 'smart contract' really a smart idea? Insights from a legal perspective". In: *Computer law & security review* 33.6 (2017).

[21] C. Heesen, V. Homburg, and M. Offereins. "An agent view on law". In: *Artificial Intelligence and Law* 5.4 (1997).

[22] M. Hepp. *GoodRelations and schema.org.* URL: http://wiki.goodrelations-vocabulary.org/GoodRelations_and_schema.org (visited on 12/13/2022).

[23] R. Iannella and S. Villata. *ODRL Information Model 2.2.* Recommendation. W3C, Feb. 2018. URL: https://www.w3.org/TR/odrl-model/.

[24] K. Janowicz, A. Haller, S. J. D. Cox, D. Le Phuoc, and M. Lefrançois. "SOSA: A lightweight ontology for sensors, observations, samples, and actuators". In: *Journal on Web Semantics* 56 (2019).

[25] R. Kibble. "Speech acts, commitment and multi-agent communication". In: *Computational & mathematical organization theory* 12.2 (2006).

[26] P. Koen, M. Kollenstart, J. Marino, J. Pampus, A. Turkmayali, S. Steinbuss, and A. Weiß. *Dataspace Protocol 2025-1-RC1.* Feb. 27, 2025. URL: https://eclipse-dataspace-protocol-base.github.io/DataspaceProtocol/2025-1-RC1/.

[27] E. Mansour, A. V. Sambra, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulnaga, and T. Berners-Lee. "A Demonstration of the Solid Platform for Social Web Applications". In: *Proceedings of Posters & Demos at the 25th International Conference on World Wide Web (WWW).* 2016.

[28] D. Moberg and R. Drummond. *MIME-based secure peer-to-peer business data interchange using HTTP (AS2).* RFC 4130. IETF, 2005. URL: https://www.ietf.org/rfc/rfc4130.txt.

[29] A. Sambra, H. Story, and T. Berners-Lee. *WebID 1.0.* Editor's Draft. 2014. URL: https://www.w3.org/2005/Incubator/webid/spec/identity/.

[30] M. Schoop, A. de Moor, and J. Dietz, eds. *Proceedings of the 1st International Conference on the Pragmatic Web.* Gesellschaft für Informatik (GI), 2006.

[31] J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language.* Cambridge, UK: Cambridge University Press, 1969.

[32] M. P. Singh. "The Pragmatic Web". In: *IEEE Internet Computing* 6.3 (2002).

[33] J. Snell and E. Prodromou. *Activity Vocabulary.* Recommendation. W3C, May 2017. URL: https://www.w3.org/TR/activitystreams-vocabulary/.

[34] M. Sporny, A. Guy, M. Sabadello, and D. Reed. *Decentralized Identifiers (DIDs) v1.0.* Recommendation. W3C, July 2022. URL: https://www.w3.org/TR/did-core/.

[35] UN/CEFACT, ed. *UN/EDIFACT: List of all Messages Release: 01B.* URL: https://service.unece.org/trade/untdid/d01b/trmd/trmdi2.htm.

[36] UN/CEFACT, ed. *UN/EDIFACT: Part 1: Introduction.* URL: https://unece.org/trade/uncefact/unedifact-part-1-introduction.

[37] UN/CEFACT, ed. *UN/EDIFACT: Part 2: UNCID – Chapter 4: Annex.* URL: https://unece.org/trade/uncefact/unedifact/part-2-uncid-chapter-4-annex.

[38] X. Wang, C. H.-J. Braun, A. Both, and T. Käfer. "Using Schema.org and Solid for Linked Data-based Machine-to-Machine Sales Contract Conclusion". In: *Proceedings of Posters & Demos at the 31st Web Conference (WWW).* 2022.

[39] C. Webber and J. Tallon. *ActivityPub.* Recommendation. W3C, Jan. 2018. URL: https://www.w3.org/TR/activitypub/.

[40] M. J. Wooldridge. "Semantic Issues in the Verification of Agent Communication Languages". In: *Autonomous Agents and Multi Agent Systems* 3.1 (2000).

[41] A. Zimmermann, A. Ciortea, C. Faron, E. O'Neill, and M. Poveda-Villalón. "Pody: A Solid-Based Approach to Embody Agents in Web-Based Multi-Agent-Systems". In: *Proceedings of the 11th International Workshop on Engineering Multi-Agent Systems (EMAS) at the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS).* Springer, 2023.