Content- and Context-Related Trust in Open Multi-Agent Systems using Linked Data

Valentin Siegert[0000-0001-5763-8265]

Technische Universität Chemnitz, Germany
valentin.siegert@informatik.tu-chemnitz.de

Abstract. In open multi-agent systems, linked data enables agents to communicate with each other and to gather knowledge for autonomous decision. Until now, trust is a factor for starting communications and ignores doubts about the content or context of ongoing communications. Several approaches are used to identify whom to trust and how human trust can be computationally modeled. Yet, they do not consider a change of context or of other agents' behavior at runtime. The proposed doctoral work aims to support content- and context-related trust in open multi-agent systems using linked data. Existing trust models need to be surveyed with respect to content- and context-related trust. A framework based on a fitting trust model and working with linked data must be developed to establish and dynamically refine trust relationships on the autonomous agents' point of view. This would enhance the applicability of decentralized systems without introducing central units as the history of the web demonstrates. Web engineers are hereby supported to work on a new level of abstraction using the decentralization, but not scrutinizing specific communication sequences.

Keywords: Solid, Trust, Content Trust, Multi-Agent Systems, Linked Data.

1 Introduction

The web is leveraging the decentralized internet architecture but is today centralized [1, 2]. The *walled gardens* [3] of social web applications can be gardens with freedom and user dynamism [4] but are also limiting access to the data for the creator/data subject. Different initiatives like EU's Next Generation Internet initiative (NGI)[1] or projects like Solid [1] are advocating a decentralized vision of the web. Decentralizing the web facilitates inter- and exchangeability of system parts, partners and providers. Yet, decentralization introduces trust challenges due to many potentially unknown parties.

Building applications in a decentralized web are challenging web engineers in a new way, especially according to trust questions. As the decentralization is bringing in more privacy and freedom for data [1], web engineers will have to work with a different view on data. Data can come from anywhere and thus it is highly questionable if this data is correct and harmless or wrong, misleading and even harmful [5]. Due to the big amount

[1] cf. https://ec.europa.eu/digital-single-market/en/policies/next-generation-internet

of data in the web, these trust decisions cannot be made by human experts but by autonomous agents.
These trust decisions should not be evaluated on a static trust relationship, nor on a relationship certified by any authority. As the agents should be autonomous to proceed with the decentral concept of the web, an authority would bring back the pilloried centralization. The dynamic evaluation of trust will give certain advantages about the fast-changing unknown parties, which can even change their behavior in specific contexts after being trusted in the first place. Thus, the autonomous agents should be able to work with dynamic trust relationships, which are content- and/or context-related and not dictated by another entity.

In the following three use cases with different complexity are presented, and their similarities are analyzed. The paper goes on with a description of the research objectives in 3, and a recent work in 4. It concludes in 5 with the research agenda.

2 Use Cases

Use Case: Solid. Solid is a well-known project which tries to give back control over the data to the creator/data subject by decentralizing online data storages [1] called *pods*. As everyone should be free to bring his pod to any application, Solid is separating the data from the application. Therewith Solid enables two novel business models for web applications: (1) the data management layer with pod hosting/providing, and (2) the application business itself avoiding data silos. Yet, there is no clear mechanism how to decide if certain data should be trusted by any application nor if the pod should accept new data from a specific application in/with a specific context or content.

Use Case: Smart Cities. The digitalization of cities also includes trustworthy systems in domains like energy distribution and public/personal traffic management. As such systems should ensure a trustworthy behavior and a respective comfortable usage, several autonomous decisions must be made at runtime. The autonomous agents making these decisions must be able to react to unpredictable events in their environment. It is required to observe how trustworthy each input data is and if it should be considered for the decision. Thus, they must ensure the overall trustworthy behavior at each intermediate decision. Otherwise, the system declaration as behaving trustworthy can be jeopardized by decisions, which were made on non-trusted or distrusted data.

Use Case: Goods Transportation. Within the goods transport sector, delivery logistics is a complex process with manual planning beforehand. It lacks an optimized dynamic, autonomous, secure and trustable way of conceptual linking one delivery to a carriage within a transportation system. The dynamic interchanging of goods in between carriages, the dynamic separation of one delivery in parts, the inferring remerging of one delivery and the dynamic separation of each carriage are also closely connected aspects at the goods transport. All these aspects need a special consideration upon trust mechanisms when it comes to an AI controlled logistics planning and execution. Regulating everything in detail without individual autonomous decisions will not support autonomous and dynamic transportation of goods.

Analysis of Use Cases. All three use cases have in common that they are separating two layers by introducing decentralization with autonomous decisions at runtime These two layers were considered as one, but the decentralization is acquiring the need to separate them. These two layers subsequently cause new trust challenges. Solid is e.g. separating the data layer from the application layer. In the context of smart cities, the autonomous agents' decision is separated from the outer view of the system. And at the goods transportation, the routing and delivering of goods and its actual transportation methods are separated. Without consideration of trust, the decentralization would decrease the trustworthy behavior of all use cases and respectively the security.

If the trust is evaluated only once for each participating party, all use cases would lack the possibility that context and content can change, and that the party could change its behavior after some time passed. Thus, all use cases benefit to evaluate the trust for each changed content and context. As the content and context can change between two communication parties without the participation of any authority, authority for specific trust mechanism would cause not only an undermining of the decentralization but also a shift in the point of view about trust. The trust would respectively not fit to a specific agent, but to the authority. Thus, all use cases require a framework which is useable for all agents and based on a respective trust model.

To exchange information and knowledge between all participating parties, services, and sensors, it is for all use case suitable to use linked data. As it is an approach for linked data, Solid is a respective special use case, but also the others can benefit of information flow by using linked data for the semantic description of data.

3 Research Objectives

To solve the mentioned problem for web engineers, the uncertainty about the inclusion of foreign data has to be abolished. Several entities can change their behavior in a decentralized system without perception, thus incoming content must be checked at each communication sequence. Subsequently, such checks could change the trust in another entity, which infers a trust check at starting a communication. Yet, not only when a new entity gets known but every time one is starting a new communication sequence.

All those checks further depend on the underlying context of the situation, because some requests and answers may not be that important than others. Hence, the trust will vary in playing a role. This context cannot always be defined before. Sometimes the situation changes due to outer influences, which cannot be predicted. Thus, the understanding of the context is an important aspect for solving the trust uncertainty.

The goal of this doctoral work[2] is to support content- and context-related trust in open multi-agent systems using linked data. Open multi-agent systems use many independent agents, who work commonly for a bigger purpose, e.g. pretend traffic jam in a city while ensuring for all individuals the fastest route. The open systems part describes that all agents could join and leave the system, without a further influence of the other parties, e.g. any application provider using Solid could start a new business or go broke.

[2] supervised by Prof. Dr.-Ing. Martin Gaedke, martin.gaedke@informatik.tu-chemnitz.de

To reach this objective, a respective framework has to be developed. As notifiable in Fig. 1, trust models should be used as a basis to dismantle the uncertainty in the introduced use cases but are having gaps. These gaps are demonstrated by the use cases as mentioned in the Analysis of Use Cases. A framework for the respective trust model in linked data is required to be developed, such that the agents in the use cases can use it. Thus, it will base on the trust model(s) and will be applied to the use cases for evaluation purposes. To utilize trust establishment in the correct way, the framework must solve the demonstrated gaps with a correct trust model underneath but also the requirements specified by the gaps. It is envisaged to find or create the perfect trust model, but the framework could also exchange the underlying model from scenario to scenario.

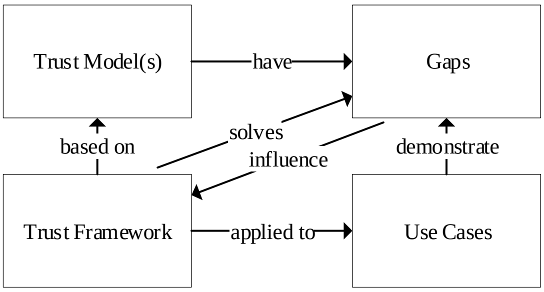

Fig. 1. Solution Concepts

4 Related Work

Policy- and reputation-based trust. Trust inferences based on policies or strict security mechanism can be grouped as *policy-based trust* [6]. Trust is here compounded “by obtaining a sufficient amount of credentials pertaining to a specific party, and applying the policies to grant that party certain access rights” [6, p. 59]. Another type of trust establishing is called *reputation-based trust* [6], where the reputation of others is used to infer trust. Thereby a *web of trust* [6] is established without any authority.

Trust models. To further compare trust values a computational trust model is required. Recent work shows that a lot of different models exist for specific scenarios [7, 8]. Cao et al. [9] introduce a model, which is very close to the mentioned use case in smart cities, where the sharing of data in such a city is modeled with regards to transparency, accountability, and privacy. Falcone and Castelfranchi [10] are “dealing with the dynamic nature of trust, and making the realization that an agent that knows he’s trusted may act differently from one who does not know his level of trust” [6, p. 65]. Besides the computational models of trust, the meaning of trust is leveraging out of social sciences, and their respective modeling of trust [11].

Content trust. Since the mentioned problem of this doctoral work requires to generate dynamic trust relationships within linked data the approach of *content trust* [12] is very important for this work. It changes the stasis of once evaluated trust relations to dynamic ones with regard to the mentioned content. But this approach is establishing a trust to another entity's content, while it lacks aspects like forgiveness, regret, distrust, mistrust and a cooperation threshold like specified by Marsh and Briggs [8].

Trust in multi-agent systems. As all three named use cases in Section 2 consider many agents in one system the interactivity between those agents also influence trust. Such multi-agent systems have respectively also to talk about the interference of others and the knowledge that others are trusting an agent in regard to the agent's behavior [13]. Huynh et al. [14] are already mentioning the uncertainty in open multi-agent systems and presenting an integrated trust and reputation model. Yet, they have issues regarding lies and do not consider a change of trust after first trust establishment.

5 Research Agenda

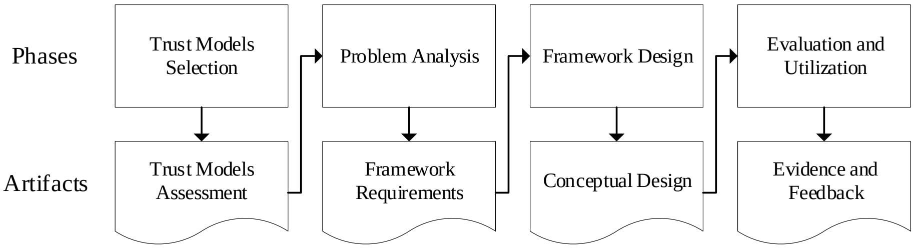

Fig. 2. Research Agenda

Trust Models Selection. At first, the correct computational trust model(s) must be found. Respective survey research about available trust models with regards to identify, analyze and evaluate them is the intended start as in Fig. 2. This survey may show the need to create a new trust model to fit in the content and context relations. But as the purpose of this doctoral work is not the creation of a new trust model, the intention is to combine available trust models. This would also benefit the model as it can use the argumentation of other models without a new work from scratch. If there is a model doing already all the work as intended, this will be used in the next steps. And as already mentioned in 3 this survey could also come to the result that several models are important and vary from use case to use case.

Problem Analysis. The framework design needs to fit the actual problem. The requirements can be found with a problem analysis with respect to the observed trust models' gaps. Some requirements are already written down within the problem statement. Yet, there could be more as the framework should be integrated into the use cases and their specific multi-agent systems.

Framework Design. With the problem analysis finished, the conceptual design of the framework can be started. A corresponding first prototype will be implemented for the utilization of the framework and its further evaluation.

Evaluation and Utilization. After the framework has a clear conceptual design with respect to its requirements, the framework is required to fit into the use cases. Therefore, the framework will be implemented and reworked by including it in each use case. The framework will hereby be included in one use case after another. Every use case utilization is thus improving the framework itself with a small evaluation in a specific scenario. After three successful use case integrations, the framework will be further evaluated with a focus on all requirements of the problem analysis.

References

1. Sambra, A.V., Mansour, E., Hawke, S., et al.: Solid: A Platform for Decentralized Social Applications Based on Linked Data, (2016).
2. Ibáñez, L.-D., Simperl, E., Gandon, F., et al.: Redecentralizing the Web with Distributed Ledgers. IEEE Intell. Syst. 32, 92–95 (2017).
3. Appelquist, D., Brickley, D., Carvahlo, M., et al.: Social Web XG Wiki. (2010).
4. Mehra, S.K.: Paradise is a Walled Garden? Trust, Antitrust and User Dynamism. 889–952 (2011).
5. Langer, A., Siegert, V., Göpfert, C., et al.: SemQuire - Assessing the Data Quality of Linked Open Data Sources Based on DQV. In: International Conference on Web Engineering. pp. 163–175. Springer, Cham (2018).
6. Artz, D., Gil, Y.: A survey of trust in computer science and the Semantic Web. J. Web Semant. 5, 58–71 (2007).
7. Golbeck, J.: Computing with Trust: Definition, Properties, and Algorithms. In: 2006 Securecomm and Workshops. pp. 1–7 (2006).
8. Marsh, S., Briggs, P.: Examining Trust, Forgiveness and Regret as Computational Concepts. In: Computing with Social Trust. pp. 9–43. Springer (2009).
9. Cao, Q.H., Khan, I., Farahbakhsh, R., et al.: A Trust Model for Data Sharing in Smart Cities. In: 2016 IEEE International Conference on Communications (ICC). pp. 1–7. IEEE (2016).
10. Falcone, R., Castelfranchi, C.: Trust Dynamics: How Trust is influenced by direct experiences and by Trust itself. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems. pp. 740–747. IEEE (2004).
11. Mcknight, D.H., Chervany, N.L.: Trust and Distrust Definitions: One Bite at a Time. In: Trust in Cyber-societies. pp. 27–54. Springer (2001).
12. Gil, Y., Artz, D.: Towards content trust of web resources. Web Semant. Sci. Serv. Agents World Wide Web. 5, 227–239 (2007).
13. Drawel, N., Qu, H., Bentahar, J., et al.: Specification and automatic verification of trust-based multi-agent systems. Futur. Gener. Comput. Syst. (2018).
14. Huynh, T.D., Jennings, N.R., Shadbolt, N.R.: FIRE: An Integrated Trust and Reputation Model for Open Multi-Agent Systems. In: Proceedings of ECAI 2004: 16th European Conference on Artificial Intelligence. pp. 18–22 (2004).