Autonomous agents for Personal Data Stores

Wout Slabbinck

IDLab, Departement of Electronics and Information Systems,
Ghent University - imec, Belgium

September 2023

1 Motivation

A lot of services nowadays are digitised. Instead of going to a shop to do groceries, you can have them delivered to your home via various platforms (Amazon, HelloFresh, etc.). This digitisation goes even further as social activities and aspects can be done online as well. For example there are a multitude of platforms that can be used to message each other (Messenger, Teams, etc.) or can be used to share photos (Instagram, Google Photos, etc.).

However, for all those services and platforms you have to make an account, where you at least fill in your first name, last name, e-mail and sometimes even more personal information. In itself, this is not a big hassle. Whenever one aspect of this personal information is changed, you will have to edit all your profiles on all platforms. Otherwise, problems can arise such as having deliveries to your old address or not receiving notifications about due payments.

Furthermore, the social activity platforms rely on your peers to also have a registered account. If that is not the case, there is no way to communicate with each other as unfortunately, you cannot share a picture from your Google Photos to Instagram (and vice versa).

A reaction to this model is Personal Data Stores (PDS). The idea is that each person has one storage where all their personal information is stored. An example of PDS is the Solid Protocol[1]. Here, data is stored in a Solid Pod, which is a personal data vault. Furthermore, applications use the protocol to interact with the data in the pod. This way, two users can use different applications that work interoperable with the data on their pods.

2 Proposal

Theoretically, the Solid Protocol solves the two problems (single source of truth and interoperability) mentioned above. Though, currently the Solid ecosystem

[1] https://solidproject.org/TR/protocol

is very application-oriented. As these are web applications, they are less suited for automated background processes like notifications and integration with third parties.

For this purpose, we see Web Agents as a solution. More specifically, for each integration or functionality, there could be a dedicated Web Agent that works in the environment of the Solid Pod of an end user and its specific functionality. As an example, we have created on such Web Agent already[2], based on the work of Kiranne, who formalised a hybrid web agent architecture [1]. Its goal is to synchronise the state Smart Home appliances with a Solid Pod. This is achieved by subscribing to a resource on the Pod and the Smart Home appliance. If a message is received from the appliance, it will be mapped to an RDF graph[3]. It will be compared to the current state and when it differs, the new state will be stored in the Solid resource. When a message is received from the Solid resource and it differs, a request will be sent according to the appliance API (or middle-ware) such that the physical device gets actuated.

This Web Agent has been made with modular components, which can be choreographed using declarative N3 rules[2]. To implement these components, the choreography engine Koreografeye[4] has been used. A declarative approach allows end-users/developers (or other agents) to focus on how the agent should behave rather than how it should be implemented.

While this Web Agent is just a demonstrator. It would be interesting to see how Web Agents for PDSs could be formalized. Such a formalisation could extend the work of Zimmerman et al. [3] and must at least provide a solution to (i) discoverability, (ii) its interface and (iii) embodiment.

With discoverability, we mean how an end-user of a PDS can discover the capabilities of the agent and/or a catalogue of agents with such capability descriptions. Furthermore, it must be clear how to initiate the agent and how additional configurations can be made. Finally, for auditory purposes and for checking the current state/configuration of the agent, it must be defined how such a Web Agent is embodied.

References

[1] Sabrina Kirrane, “Intelligent software web agents: A gap analysis,” *Journal of Web Semantics*, vol. 71, p. 100659, 2021, publisher: Elsevier.

[2] T. Berners-Lee and D. Connolly, “Notation3 (n3): A readable rdf syntax,” *W3C team submission*, vol. 28, 2011.

[3] 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-

[2] https://github.com/SolidLabResearch/Solid-Agent/blob/main/documentation/iot/README.md

[3] https://www.w3.org/RDF/

[4] https://github.com/eyereasoner/Koreografeye

Agent-Systems," in *11th International Workshop on Engineering Multi-Agent Systems (EMAS2023)*, 2023.