Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220462

Automating the Response to GDPR's Right of Access

Beatriz ESTEVES [a,1], Víctor RODRÍGUEZ-DONCEL [a], Ricardo LONGARES [a]

[a] *Ontology Engineering Group, Universidad Politécnica de Madrid, Spain*

Abstract.
With the enforcement of the European Union's General Data Protection Regulation, users of Web services – the 'data subjects' –, which are powered by the intensive usage of personal data, have seen their rights be incremented, and the same can be said about the obligations imposed on the 'data controllers' responsible for these services. In particular, the 'Right of Access', which gives users the option to obtain a copy of their personal data as well as relevant details such as the categories of personal data being processed or the purposes and duration of said processing, is putting increasing pressure on controllers as their execution often requires a manual response effort, and the wait time is negatively affecting the data subjects. In this context, the main goal of this work is the development of an API, which builds on the previously mentioned structured information, to assist controllers in the automation of replies to right of access requests. The implemented API method is then used in the implementation of a Solid application whose main goal is to assist users in exercising their right of access to data stored in Solid Pods.

Keywords. digital rights management, GDPR, right of access, Solid

1. Introduction

With the enforcement of the European Union's General Data Protection Regulation (GDPR), users of Web services have seen their rights as GDPR 'data subjects' being expanded when it comes to the processing of their personal data. On the other hand, on top of other GDPR-related obligations, 'data controllers', the entities that effectively process the data, have seen an increase in workload related to the response to data subject's right-related requests. GDPR's Chapter III[2] details a set of 10 data subject rights, starting with the 'Right to be Informed' described in Articles 13 and 14 and ending with the 'Right to object to automated decision making' in Article 22. Considering this, data controllers would benefit from having the information they need to provide to data subjects in a structured format to automate the response to such requests [1]. In particular, the 'Right of

[1]Corresponding author: beatriz.gesteves@upm.es
[2]https://gdpr-info.eu/chapter-3/

Access'[3] is putting more and more pressure on controllers as they not only have to provide the purpose for which the data is being used or the types of data being processed but also need to provide a copy of said data. As this task is usually done manually, the wait time can negatively affect data subjects.

In addition, with the emergence of decentralised data storage solutions, such as Solid[4], as an alternative to the traditional centralised data silos, new challenges appear as the data subject–data controller roles are still not adequately defined in this decentralised contexts. In this context, the creation of Application Programming Interface (API) services would help with the automation of right-related requests as, at their core, they are a 'request–response' type of software interface and consequently can be used in different types of software.

Taking this into consideration, this contribution will focus on the following research objectives:

RO1. Implementing an API method that automates the reply to an access right request, making use of RDF information.
RO2. Developing a Solid application which uses the implemented method to assist users in exercising their right of access in a decentralised storage environment, such as data stored in Solid Pods.

This paper is organized as follows: Section 2 introduces the GDPR's Data Subject Access Right and its core requirements, Section 3 discusses related work in this area, Section 4 describes the implementation of an API method that automates the response to an access right request and a Solid application that uses said API method to assist users in exercising their right of access to data stored in Solid Pods and Section 5 presents conclusions and future lines of work. Online supplementary material is provided at https://protect.oeg.fi.upm.es/access-right/, including a demonstration of the implemented Solid application.

2. The Right of Access

In an attempt to provide users with more transparency on how their personal data is being processed, GDPR's right of access puts emphasis not only on the "right to obtain a copy" of said data but also an additional set of information pieces needs to be provided to the subject related with the context in which the processing is made. Therefore, the right of access's main goal is to give people sufficient and clear information about how their personal data is processed so that subjects know that their data is being handled according to their expectations and its quality is being verified – this will then facilitate the exercise of other rights, such as the right to be forgotten or to rectification. Data subjects do not have to provide a justification to request access to the data nor do they need to pay a fee to exercise such right unless further copies of the same data are requested. Unless it is obvious that the request is being made in violation of other regulations, the data controller has the duty to reply to the data subject's request. Moreover, in addition to confirming whether or not they are processing personal data and

[3] https://gdpr-info.eu/art-15-gdpr/
[4] https://solidproject.org/

providing a copy of said data, controllers have a duty to provide information regarding the purposes of the processing, the categories of personal data being processed, the recipients or categories of recipients of said data, the duration of the processing, the existence of other data subject rights, including the right to lodge a complaint with a supervisory authority, the source of data if not collected directly from the data subject and the existence of automated decision-making. In addition to the requirements specified in GDPR, on January 18th 2022, the European Data Protection Board (EDPB) issued a set of guidelines specifically on the right of access [2].

3. Related Work

There are a few API implementations specifically targeting the response to data subject right-related requests, however, they only focus on providing a copy of the personal data and leave out all the other requirements specified by the GDPR.

Microsoft Graph [3] is a platform developed to access Microsoft 365 data. In particular, the Microsoft Graph compliance and privacy APIs [4] were developed to create and manage data subject access requests and to help developers and enterprises to easily identify data subjects and find their personal information. Although the process through which the data is obtained is automated, several components require manual intervention, such as confirmation as to whether the data is being processed.

Oracle's Data Privacy API focus on providing a solution for enterprises that use Oracle databases to store personal data [5]. Currently, it supports the implementation of two types of requests, the already discussed Right of Access and also the Right to be forgotten – a right that can be exercised by a data subject when they want the system in question to erase all data related to them.

AppsFlyer [6] goes one step further by providing an API that not only supports access and erasure requests but also deals with the Right to Data Portability – transfer data from one controller to another – and the Right to Rectification – correct inaccurate data. The request flow in AppsFlyer also involves manual intervention, as when a data subject submits a request, the app owner has to forward the request to AppsFlyer.

Through the analysis of these solutions, it is possible to conclude that there is a gap related to the implementation of an API service that fulfils all the requirements of a GDPR Right of Access request since all solutions only focus on providing a copy of the data and don't provide detailed information regarding purposes for processing, duration of the processing and so on.

4. Implementation

4.1. Research Methodology

The used methodology approach encompassed the following steps:

1. An evaluation of current gaps on the right of access APIs was performed.

2. Similar regulation from other jurisdictions was reviewed in order to understand if new requirements needed to be added into consideration.
3. Semantic Web vocabularies were used to tag the data in terms of the personal data they contain, to specify the policies that determine the access to said data and to store the consent record of an authorized access request.
4. The API method and documentation were developed.
5. Solid's personal data storage ecosystem was then chosen to verify the applicability of the API method as it is based on Web standards.

Further information on steps 1 to 3 is provided at https://protect.oeg.fi.upm.es/access-right/.

4.2. API development

The main technologies used to implement the API were the expressjs[5] and swagger-ui-express[6] libraries, used to develop the API and create its respective documentation. Inrupt's JavaScript client libraries[7] were also used to authenticate the user and to handle data stored in a Solid Pod.

Initially, the developed API only had one parameter which was related to the identity of the user as this is the only requirement described by the GDPR for a data subject to be allowed to exercise their right of access. However, since the data subject may, for example, be interested in accessing only certain categories of data or only accessing data used for a certain purpose, data categories and purposes were added as request parameters to allow the data subject to have a more fine-grained access right.

The main function of the API is to obtain the data stored in the Solid Pod and send it to the user as a JSON file with two components – a boolean variable that will be true in case personal data that matches the request is found in the Pod and a JSON object that contains the respective list of found resources. To enable access to the data, the user must be logged into their Pod. As previously stated, the authentication protocol is implemented using Inrupt's libraries – a session is generated and stored so that information regarding the identity of the user (present in a WebID profile document in the case of Solid) can be passed to the API request.

Once the user is logged in, the API can process an access request. Initially, the URI of all resources stored in the Pod is collected and matched with the request. If the request is made without specifying any further parameters, then all the resources present in the Pod will be returned independently of the categories of personal data that they include or the purpose for which its processing is allowed. If a specific set of personal data categories is specified along with the access request then only those categories will be returned. However, it must be noted that for this feature to work, the resources in the Pod need to include a RDF statement, using for instance the Extended Personal Data concepts for DPV[8], to specify which type of data they contain. Finally, in case the user only wants to

[5] https://expressjs.com/es/
[6] https://www.npmjs.com/package/swagger-ui-express
[7] https://docs.inrupt.com/developer-tools/javascript/client-libraries/
[8] https://w3id.org/dpv/dpv-pd

access data that is being used for a particular purpose, access control policies that define the purpose for processing the stored resources need to be defined and kept in their Pod. For the purposes of this work, we assume that users are using the Open Digital Rights Language (ODRL) Profile for Access Control (OAC) [7] to create policies to govern the access to their Solid-stored data[9]. Using these policies the API method matches the request purpose with the stored policies' purposes and if there is a match then the corresponding data is returned. In addition to the policies, we assume that consent records, corresponding to granted data access requests, are kept in the Pod. The generation and modelling of consent records are based on previous work [8]. These records are then used by the API to retrieve the entities that accessed the data.

A public repository with the developed code is accessible at https://github.com/besteves4/access-right-api.

4.3. Exercising the Right of Access to Solid Pod data

As previously stated, Solid is a protocol focused on providing its users with decentralised personal data storage. Currently, access control to Solid-stored resources is specified using the Web Access Control specification which uses Access Control Lists (ACLs)[10] to define which agents have access to Solid resources. However, these ACL authorisations don't allow the specification of purposes for the access to the resources as well as not permitting the definition of specific access to particular types of personal data. In this context, as specified in the previous section, this work makes use of OAC policies to overcome this issue. OAC[11] uses the Data Privacy Vocabulary (DPV), which provides taxonomies for the specification of relevant privacy and data protection information, and ODRL, which allows for the expression of rich policies over digital assets.

As the users need to be able to select specific types of personal data and/or specific purposes for the processing of said data to have a more fine-grained right of access, the developed Solid application includes two drop-down trees that use DPV's personal data categories and purposes taxonomies to populate its structure. The selected categories are then used to feed the API request call. For each returned resource, the URI is provided, as well as the category of personal data included in the file, the agents that accessed the data and a list of the policies governing the access to said resource. A download button is also available so that the user can obtain a copy of the resource data. A demonstration of the developed Solid application is available at https://protect.oeg.fi.upm.es/access-right/ and the public repository of the code is accessible at https://github.com/besteves4/access-right-solid for further development.

This solution provides an advance in relation to the state-of-the-art reviewed solutions as it provides granular information on the personal data categories contained in the resources, as well as the purpose for which it was/can be used, in addition to the provision of a copy of the data.

[9]The SOPE application available at https://github.com/besteves4/solid-sope can be used to automatically generate these policies without having knowledge on ODRL and its semantics.

[10]http://www.w3.org/ns/auth/acl#

[11]https://w3id.org/oac

5. Conclusions

This work explored the implementation of an API service that can be used for the automation of GDPR's Right of Access. Current state-of-the-art solutions developed to assist data subjects and data controllers in the exercising and resolution of data subject right-related requests focus on providing users with the 'right to obtain a copy' of the data but do not fulfil all the requirements set by the GDPR. Furthermore, these APIs are currently not equipped to deal with requests regarding data stored in decentralised systems such as Solid Pods.

Therefore, the main contribution of this work relies on the development of an open-source API that can be used in the context of decentralised systems to provide data subjects with a 'fine-grained' right of access where it can be explicitly checked which data is being used for what purpose in addition to obtaining a copy of the data.

In future lines of work, the API can be improved to provide a more complete answer to a Right of Access request – information about other data subject rights should be provided, as well as clear information regarding recipients of the data, including identity and contact information. Furthermore, audit logs of the process of exercising the right should be kept in a dedicated container in the Pod, for future inspection. Moreover, there are factors that have not been considered for the sake of practicality. For example, we have assumed that during the period of time in which the resources are stored in the Pod under the effects of a policy, they can be automatically provided through the API. This feature can also be extended to deal with new policy constraints, such as a limited time duration for the storage or a periodicity constraint. Also, new parameters for filtering the requested data could be added to the API and to the Solid application.

Funding Acknowledgments This research has been supported by European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813497 (PROTECT).

References

[1] Esteves B, Rodríguez-Doncel V. Analysis of Ontologies and Policy Languages to Represent Information Flows in GDPR. Semantic Web Journal. 2022.
[2] EDPB. Guidelines 01/2022 on data subject rights - Right of access - Version 1.0; 2022. Available from: https://bit.ly/3sgtnSd.
[3] Overview of Microsoft Graph; 2022. Available from: https://bit.ly/3DhPnCp.
[4] Use the Microsoft Graph compliance and privacy APIs; 2022. Available from: https://bit.ly/3MORtwZ.
[5] Appendix B: Using the Data Privacy API; 2021. Available from: https://bit.ly/3MPJFLn.
[6] Implementing the OpenGDPR API; 2022. Available from: https://bit.ly/3eSFphG.
[7] Esteves B, Pandit HJ, Rodríguez-Doncel V. ODRL Profile for Expressing Consent through Granular Access Control Policies in Solid. In: 2021 IEEE EuroS&PW; 2021. p. 298-306.
[8] Esteves B, Rodríguez-Doncel V, Pandit HJ, Mondada N, McBennett P. Using the ODRL Profile for Access Control for Solid Pod Resource Governance. In: Groth P, Rula A, Schneider J, Tiddi I, Simperl E, Alexopoulos P, et al., editors. The Semantic Web: ESWC 2022 Satellite Events; 2022. p. 16-20.