ODRL Profile for Expressing Consent through Granular Access Control Policies in Solid

Beatriz Esteves
Ontology Engineering Group
Universidad Politécnica de Madrid
Madrid, Spain
beatriz.gesteves@upm.es

Harshvardhan J. Pandit
ADAPT Centre
Trinity College Dublin
Dublin, Ireland
pandith@tcd.ie

Víctor Rodríguez-Doncel
Ontology Engineering Group
Universidad Politécnica de Madrid
Madrid, Spain
vrodriguez@fi.upm.es

Abstract

Solid, the emerging technology for organizing data in decentralized stores, relies on a simple authorization mechanism for granting access to data. Solid's personal online datastores (Pods) are ideal for keeping personal data, as they allow individuals to represent the access permissions in a very simple manner using Access Control Language (ACL) expressions. Whereas these expressions suffice for yes/no and read/write permissions, they cannot represent more complex rules nor invoke regulation-specific concepts. This paper describes an extension of the ACL language and algorithm to implement consent and data requests. The extension is based on the Open Digital Rights Language (ODRL) policy language, which allows expressing rich rules, and the Data Privacy Vocabulary (DPV), which permits invoking privacy and data protection-specific terms. Some usage examples illustrate this proposal.

Index Terms

access control, consent, data protection, decentralized datastores, privacy, DPV, GDPR, regulatory compliance

Accepted for publication and presentation at International Workshop on Consent Management in Online Services, Networks and Things (COnSeNT 2021) co-located with 6th IEEE European Symposium on Security and Privacy (EuroS&P). To published in IEEE Xplore Conference Proceedings: https://ieeexplore.ieee.org/xpl/conhome/1813044/all-proceedings

I. Introduction

Solid is a specification[1] for decentralised personal data stores (called 'Pods') based on the tenets of user control and interoperability using linked data and web standards. Access to data stored within Pods is governed by the access control specification Web Access Control[2] (WAC) which uses Internationalized Resource Identifiers (IRIs) to represent resources and agents, and stores authorisation statements within an Access Control List (ACL) defined per resource or inherited from the parent resource within the IRI. Currently, WAC supports four operations: read, write, append, and control (of ACLs) which can be declared for specific agents or trusted apps, and is interpreted as a prohibition by default unless there is an ACL authorisation permitting it. As per Solid, the user controlling the ACL is the entity responsible for deciding who has access to the data. In this manner, Solid intends to provide and standardise implementations for users to store their own data and have control over who they wish to share it with.

Given that Solid provides a way for personal data to be stored and managed by individuals, it is important to consider the impact and application of data protection and privacy laws such as the European General Data Protection Regulation[3] (GDPR) that defines specific concepts and obligations for how personal data can be collected, stored, used, and shared. In addition, such laws also specify requirements for provision and validity of legal bases such as consent used to justify processing of data. While we focus on GDPR in this article, the argument applies to other existing and emerging laws following similar trends[4].

GDPR requires controllers to provide information such as identity, purpose, personal data categories, legal bases, and recipients where they collect personal data directly (Art.13) or indirectly (Art.14) from individuals. The intent behind the provision of this information is to provide transparency and accountability, and to benefit the individual in making informed decisions regarding use of their personal data. The information is made available conventionally through notices, consent dialogues, and privacy policies; and is also relevant for the utilisation of technology in compliance-related tasks.

Applied to Solid, this information can be presented using conventional methods, e.g., showing a notice provided on a website controlled by the data requester, and the resulting authorisation stored within the ACL. However, we argue that empowering individuals to control their data practices requires the Solid Pod to contain this information as well so that the individual has the opportunity to: (i) introspect use of their personal data within an environment under their control; (ii) store consent and authorisations for accountability purposes; (iii) determine their data sharing preferences; and (iv) be assisted in expressing and

[1] https://solidproject.org/TR/protocol
[2] https://solid.github.io/web-access-control-spec/
[3] https://eur-lex.europa.eu/eli/reg/2016/679/
[4] Global privacy laws catalogue http://www.worldlii.org/int/special/privacy/

enforcing their data sharing preferences, including withdrawal of consent. Achieving this requires understanding: (a) what the law says/requires in terms of information to be provided and its use for exercising a legal basis such as consent; (b) what do individuals need to know or would like to know; and (c) what forms of control would the individuals like to have in the context of their personal data.

Through this paper, we address the above goals with the research question, ***How can Solid's ACL be extended to specify and enforce an individual's data sharing preferences?***. We generalise *consent* to *data sharing preferences* to align consent requests with authorisation permissions within the Solid architecture, and later discuss the requirements for consent regarding information, explicitness, and withdrawal. For this, we first propose extending the ACL mechanism by utilising Open Digital Rights Language (ODRL) policies that express the permissions or prohibitions associated with data stored in a Solid Pod and utilise the Data Privacy Vocabulary (DPV) as a controlled vocabulary for representing metadata relevant to the legal compliance. Along with the description and demonstration of its practical applicability, we also discuss how this mechanism can be utilised for consent requests, the potential of automation in the process, and the challenges and issues for successful implementation of our solution.

Section II outlines the motivations and the resulting requirements we used to inform our approach; Section III provides an overview of related work; Section IV describes the proposed ODRL profile; Section V provides details of a prototype demonstration; Section VI presents a discussion on challenges and issues and Section VII concludes the paper.

II. Rationale

This section outlines the rationale and resulting requirements motivating the need for our proposed approach in extending Solid's existing ACL with additional functionality. The motivation for adopting the technologies we propose can be substantiated in the following points:

1. Legal compliance - organisations and individuals wish to:
   a) Document activities and provenance regarding personal data processing, requests for access, notices and use of logs, e.g., for audits;
   b) Determine applicable rights, obligations, and requirements based on jurisdictional laws or contextually, e.g., specific categories of personal data;
   c) Evaluate, assess, or validate if obligations and requirements are fulfilled;
   d) Implement security, specifically access to data.
2. User-defined preferences for data sharing and use
   a) Express human-centric preferences, e.g., willingness to share particular data for research, or prohibition of profiling and surveillance;
   b) Granularity for specifying broad permissions, e.g., permit data use for scientific research, or prohibit any third party data collection;
   c) Granularity for specifying narrow permissions, e.g., permit sharing contact details for a specific app, or prohibit accessing a confidential resource;
   d) Conflict resolution going from local to global, e.g., generally prohibiting sharing of location, but making an exception for specific services.
3. Transparency of data practices
   a) Individuals want to know who is using which data categories, for what purposes, sharing it with whom, and under what legal basis;
   b) Individuals want to understand permissions/authorisations they already have provided based on self-defined context, e.g., for specific services, or based on purposes, or categories of personal data;
   c) Organisations may want to specify machine-readable data policies accessible by users.

A. Requirements

R1. Support specifying user preferences as policies.
R2. Incorporate vocabulary specifying/aligned to law.
R3. Support permissions/prohibitions at arbitrary granularity.
R4. Support identifying and resolving conflicts with scope.
R5. Record (store) policies used to authorise access.
R6. Support querying/analysis of policies and authorisations.

The existing functionality of Solid's WAC partially implements requirements R1, R3, and R5 by enabling users to declare granular policies (e.g., for specific agents or groups of agents) that are stored in the Pod; and although the ACL can specify permissions, it is currently not possible to model prohibitions. For continued interoperability and adherence to the specification,

the proposed extension to Solid's ACL must ideally continue to implement existing functionality while incorporating the legal and user-centric requirements.

The requirements R1 to R3 can be satisfied with an extension of the ODRL ontology as an ODRL profile. In order to determine the extent of such profile, formal competency questions were made using the methodology described by Suárez de Figueroa [1] and shown in Table I.

TABLE I
ONTOLOGY REQUIREMENT SPECIFICATION DOCUMENT

ODRL Profile for Access Control in Solid	
1. Purpose	
The purpose of this profile of ODRL is to support policies determining the access control to personal data stored in Solid Pods	
2. Scope	
The scope of this profile is limited to the definition of an ODRL Profile for Access Control in Solid. In particular, the introduced elements will serve one of these purposes: (i) define actions supporting enforcement of current ACL verbs, (ii) define data protection-related actions and restrictions defined in GDPR, (iii) any vocabulary element to support policy patterns that can be anticipated to be common, and (iv) elements necessary to support the authorization reasoning decision.	
3. Implementation Language	
OWL	
4. Intended End-Users	
Developers of Solid servers and Solid clients.	
5. Intended Uses	
Use 1. Declaration of a policy by an individual storing personal data in a Pod Use 2. Request of data made by a person or application to gain access to the data in different modalities Use 3. Contextual elements to be considered in the authorization decision. Use 4. Explanation of the authorization decision .	
6. Ontology Requirements	
a. Non-Functional Requirements	
NFR 1. The ontology shall be published online with standard documentation.	
b. Functional Requirements: Groups of Competency Questions	
CQG1. Related to authorization	CQG2. Related to GDPR
CQ1. Which actions are to be authorized? CQ2. Which requirements are to be authorized? CQ3. Who are the parties intervening in policy? CQ4. Which is the priority of a certain policy? CQ5. Which are the contextual elements to be considered in the authorization decision?	CQ6. Which obligations and requirements, and information about personal data and its processing are necessary? CQ7. Which is the legal identification of the policy parties?

III. BACKGROUND AND RELATED WORK

A large body of work exists within the general fields of 'business process compliance' [2] and 'legal ontologies' [3] which specify how jurisdictional laws translate into specific obligations and requirements for systems. For this work, we focus on the existing research and solutions limited to addressing requirements presented in Section II, and present the state of the art across two broad areas of: (i) access control policies using linked data, and (ii) specifying information about personal data and metadata processing.

A. Access Control Policies using Linked Data

Access control in the context of linked data and RDF has a plethora of models, approaches, and policy specification languages that can be used to express and evaluate rules for access to resources. The survey paper by Kirrane et al. [4] provides an overview of the state of the art and outlines relevant standards and their adoption. Of these, WebID and WAC, as standards, are already used by Solid.

XACML[5] (eXtensible Access Control Markup Language) is a standard specifying an access control policy language, an architecture, and a processing model for evaluating access control rules in policies. Its use has been demonstrated for GDPR regarding consent [5] and access control [6].

The Open Digital Rights Language[6] (ODRL) [7], [8], a W3C Recommendation, is a policy expression language used to represent permissions and prohibitions in terms of actions over assets that can be further restricted using constraints and duties, and which supports 'profiles' extensions[7]. Existing work has used the profiles mechanism to represent obligations and compliance requirements for GDPR using ODRL. Two noteworthy approaches for this include that by Agarwal et al. [9] for an ODRL profile modelling legislative rights and obligations in GDPR with support for multiple legislations; and the work by De Vos et al. [10] who also define an ODRL profile to model GDPR compliance requirements that are translated into ASP rules for automatic compliance checking.

[5]https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
[6]https://www.w3.org/TR/odrl-model/
[7]ODRL Profile Best Practices - https://w3c.github.io/odrl/profile-bp/

Though XACML provides an arguably richer set of expressivity and a reference implementation for access control, ODRL profiles provide a convenient extension mechanism and integration with existing RDF-based systems within the Solid architecture. For this work, we therefore chose ODRL for the ease of adapting it to our requirements.

B. Specifying Personal Data and Processing Metadata

The SPECIAL[8] project has produced open-access vocabularies and implementations for expressing, recording, and validating user policies with data requests regarding consent and data handling [11], [12]. The expressiveness of SPECIAL's policy language (SPL) is minimal: it consists of purpose, processing operations, personal data categories, data storage location and duration, and recipients, which are then used with a compliance reasoner to match user and requester policies for consent.

The application of SPL to Solid's ACL mechanisms has been proposed by Havur et al. [13] which meets several requirements outlined in Section II, and also describes evaluation of policies for compliance based on consent through utilisation of SPECIAL's compliance reasoner and framework. It thus provides a guiding direction for our approach, and although it does not specify details about implementing granularity or expressing prohibitions, it does mention the suitability of using ODRL in its discussion of future work.

The MIREL[9] project has produced the PrOnto [14] ontology which provides concepts for modeling relationships between privacy agents, data types, processing operations and deontic specifications to support compliance with the GDPR. DAPRECO is a repository of rules written in LegalRuleML [15] that represents the provisions of the General Data Protection Regulation based on PrOnto. The BPR4GDPR[10] project is an ongoing effort involving the Information Model Ontology [16] that specifies activities and entities in organisations, processing operations, events, and purposes of data handling which are used by the Policy Model Language [17] to define deontic policies for data access and control.

GDPRtEXT[11] [18] provides a vocabulary of concepts associated with GDPR as a thesaurus. GConsent[12] [19] is an ontology for representing information about consent as an entity with a life-cycle based on GDPR by modeling state, entities, purposes, personal data categories, and provenance. DUO[13] (Data Use Ontology) enables modeling restrictions based on organisations, purposes, and data categories for using medical data for research, but does not include legal concepts.

TABLE II
MAIN ELEMENTS IN THE PROFILE

Profile term	Instance of	Comment
oac:Access	odrl:Action, acl:Access	operations for resource access
oac:Processing	odrl:Action, dpv:Processing	processing of personal data
oac:PersonalDataCategory	odrl:Asset, dpv:PersonalDataCategory	categories of personal data
oac:Purpose	odrl:LeftOperand, dpv:Purpose	purposes for personal data processing
oac:Recipient	odrl:LeftOperand, dpv:Recipient	entities receiving personal data
oac:LegalEntity	odrl:Party, dpv:LegalEntity	legally defined, e.g., data controllers

The Data Privacy Vocabulary[14] (DPV) [20] is an outcome of the W3C Data Privacy Vocabularies and Controls Community Group[15] (DPVCG), established within the SPECIAL project to extend its work through community participation. DPV provides top-down taxonomies that are based on GDPR but intended to be jurisdiction-agnostic to represent personal data handling practices in terms of personal data categories, purposes, processing, technical and organisational measures, legal entities, legal bases, rights, and risks.

The work proposed by Havur et al. [13] describing use of SPL in Solid's ACL provides a strong motivation for utilising SPECIAL's language and tools. However, we chose DPV given that it is open and accessible, is an extension of SPL, and is more comprehensive.

IV. AN ODRL PROFILE FOR SOLID'S ACL

From the existing work, we decided to create an ODRL profile using DPV as a controlled vocabulary for expressing and implementing user-preferences as profiles in Solid's existing access control system which uses the Web Access Control (WAC) specification for authorisation.

While ACL authorizations suffice for establishment of read/write permissions, they cannot be used to specify prohibitions or obligations over the resources and cannot define more complex rules. Furthermore, they cannot be used to invoke legal

[8] https://www.specialprivacy.eu/about/project-overview
[9] https://www.mirelproject.eu/index.html
[10] https://www.bpr4gdpr.eu/
[11] https://w3id.org/GDPRtEXT/
[12] https://w3id.org/GConsent
[13] http://www.obofoundry.org/ontology/duo.html
[14] https://w3.org/ns/dpv
[15] https://www.w3.org/community/dpvcg/

concepts related to data protection and privacy. To overcome these issues, we extend the ACL with ODRL policies that specify permissions and prohibitions and utilise DPV to specify and invoke data protection and privacy concepts.

We use the following prefixes and namespaces in this paper:

Prefix	Namespace
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
cert	http://www.w3.org/ns/auth/cert#
xsd	http://www.w3.org/2001/XMLSchema#
odrl	http://www.w3.org/ns/odrl/2/
dpv	http://www.w3.org/ns/dpv#
acl	http://www.w3.org/ns/auth/acl#
oac	https://w3id.org/oac/

In order to use ODRL and DPV with WAC/ACL, our profile specifies alignments between the vocabularies so as to permit their semantics to be interpreted correctly. Table II lists terms in our ODRL profile along with information regarding the alignment with ODRL, DPV and ACL and their interpretation. The profile can be accessed at https://w3id.org/oac/.

We decided to focus on *Purpose, Personal Data Category, Processing,* and *Recipients* as the minimum 'core concepts' for our profile, and (for now) leave out other DPV concepts such as technical and organisational measures, rights, and risks. As DPV's processing concepts (e.g., use, store, share) differ from their WAC counterparts (read, write, append, control), we mapped the two based on our interpretation of possible WAC operations permitted under each processing type. Under our interpretation, `acl:Read` corresponds to `dpv:Use, dpv:Collect`, and `acl:Write` corresponds to `dpv:Store, dpv:MakeAvailable`. Of note in this exercise is that concepts such as 'share' and 'transfer' have no exact corresponding concept in WAC, which calls for further introspection in aligning legal processing concepts with access control operations.

The ODRL profile relies on invocation of legal concepts using DPV, which in turn relies on declarative metadata for specifying which resource within the Pod contains what categories of personal data. To alleviate this, we presume that the ACL for a resource also declares its personal data category, if any. Absence of explicit definition is interpreted as the resource being generically personal data represented by `dpv:PersonalData`. Therefore, an ACL specifying a policy involving `dpv:Contact` indicates that the resource contains contact information.

Using our ODRL profile, it is possible to express preferences in terms of granular (broad or fine-grained) policies that outline permissions and prohibitions based on user-specific contexts such as for personal data categories, purposes, or recipients. For the examples in this document, we consider Anne as identified by her WebID <https :// anne.databox.me/ profile /card#me>.

First, we provide examples - Listing 1 and Listing 2 are two user preferences expressed within the ACL with our ODRL profile. In Listing 1, a broad permission over contact data is set by the owner of the Pod for the purpose of research and development, which permits read access operations over her contact data for the purpose of research and development.

Listing 1. Read-only policy for Contact for R&D Purposes

```
:policy-1 a odrl:Policy ;
    odrl:profile oac: ;
    odrl:permission [
        a odrl:Permission ;
        odrl:assigner :anne ;
        odrl:target oac:Contact ;
        odrl:action oac:Read ;
        odrl:constraint [
            odrl:leftOperand oac:Purpose ;
            odrl:operator odrl:isA ;
            odrl:rightOperand
                dpv:ResearchAndDevelopment ] ] .

:anne a oac:DataSubject ;
    cert:key <https://anne.databox.me/profile/card#me> .
```

Listing 2. Prohibition to share resource with more than 3 third parties

```
:policy-2 a odrl:Policy ;
    odrl:profile oac: ;
    odrl:prohibition [
        a odrl:Prohibition ;
        odrl:assigner :anne ;
        odrl:target <https://anne.databox.me/docs/file1> ;
        odrl:action oac:Share ;
        odrl:constraint [
            odrl:leftOperand oac:Recipient ;
            odrl:operator odrl:gt ;
            odrl:rightOperand "3"^^xsd:integer ] ] .
```

The second set of examples concern requests made by services to access personal data stored in users' Pods. In Listing 3, an app requests permission to use and store the email address and social network information of users for the purpose of registering and authenticating to their service.

Listing 3. Request to use data for Registration purposes

```
:app-1 a odrl:Policy ;
    odrl:profile oac: ;
    odrl:permission [
        a odrl:Permission ;
        odrl:assignee :app-1-controller ;
        odrl:target oac:EmailAddress, oac:SocialNetwork ;
        odrl:action oac:Use, oac:Store ;
        odrl:constraint [
            odrl:leftOperand oac:Purpose ;
            odrl:operator odrl:isA ;
            odrl:rightOperand
                dpv:RegistrationAuthentication ] ] .

:app-1-controller a oac:DataController ;
    cert:key <https://example-app-1.com> .
```

In Listing 4, an app requests permission to collect and produce a copy of (sensitive) health records, medical prescriptions and health history of users for academic research and to publish it in anonymised forms.

Listing 4. Request to collect and share anonymised Health Records

```
:app-2 a odrl:Policy ;
    odrl:profile oac: ;
    odrl:assignee :app-2-controller ;
    odrl:target oac:HealthRecord,
        oac:Prescription, oac:HealthHistory ;
    odrl:permission [
        a odrl:Permission ;
        odrl:action oac:Collect, oac:Copy ;
        odrl:constraint [
            odrl:leftOperand oac:Purpose ;
            odrl:operator odrl:isA ;
            odrl:rightOperand dpv:AcademicResearch ] ] ;
    odrl:permission [
        a odrl:Permission ;
        odrl:action oac:Anonymise, oac:MakeAvailable ;
        odrl:constraint [
            odrl:leftOperand oac:Recipient ;
            odrl:operator odrl:isA ;
            odrl:rightOperand dpv:ThirdParty ] ] .

:app-2-controller a oac:DataController ;
    cert:key <https://example-app-2.com> .
```

These examples, as well as the profile serialisation, can be accessed at https://github.com/besteves4/odrl-access-control-profile.

For purposes of accountability and transparency, the request is stored within the ACL of the resource so that it can be accessed after authorisation. Anne can then utilise, with the support of her Solid Pod, simple SPARQL queries to inquire who is using her data and for what purposes - based on utilising the stored requests and authorizations.

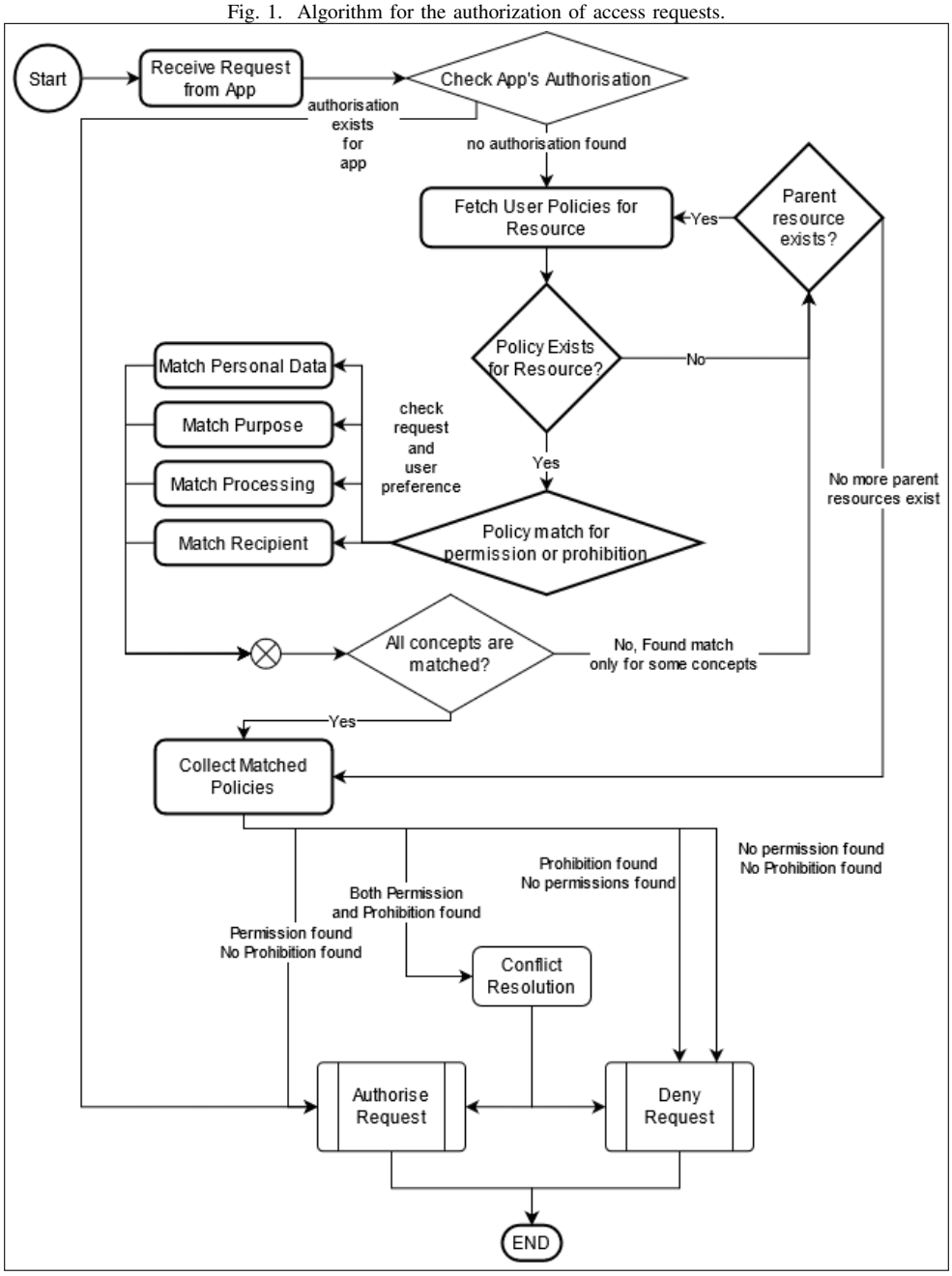
V. Implementation & Architecture

In this section, we describe the implementation of an algorithm for matching and authorizing access requests. A prototype implementation of the matching algorithm is available at https://protect.oeg.fi.upm.es/solid-consent/. In Fig. 1, a flowchart for the algorithm is presented.

After fetching the policies of the app which is making a request and the Pod user's permissions and prohibitions, the authorization mechanism should first check if there is an explicit authorization for this particular app. If so, then the request must be authorized, otherwise each app permission request should be matched with the user policy for the resource being requested. In the case where a policy is found for the specific resource, the algorithm proceeds with matching the request and the user preference in relation to the personal data, purposes, processing and recipients presented in the policies. If no match is found for any of the categories, then the algorithm must look for the parent resource and its respective policy and repeat this process until either a match is found for all categories or until there is no more parent resources, i.e. Pod-level resources.

Once every policy is checked and matched, the algorithm has to reason over the found permitted and prohibited requests and determine the authorized access requests based on the conflict resolution mechanism.

Fig. 1. Algorithm for the authorization of access requests.

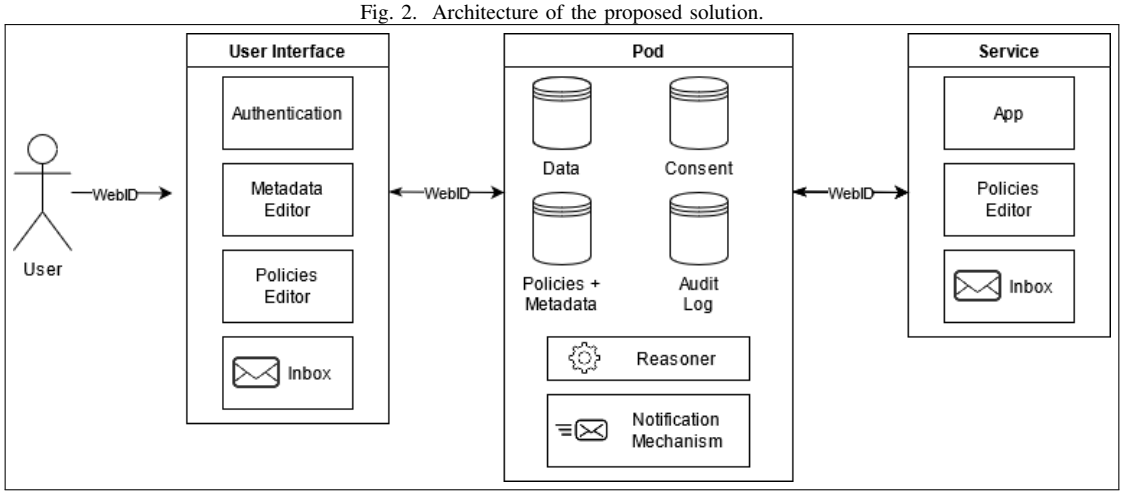


The proposed Pod architecture, visualised in Fig. 2, is an extension to the existing Solid Pod specification and implementations. We propose to add a Consent datastore component to keep a record of the consent actions already given, associated with a copy of the accepted app requests, as well as an Audit Log component to store metadata related to logins, access requests, changes in policies and consent authorisations and so on. In this context, the permissions and prohibitions specified by the users or the contents of the policy matching, described above by the algorithm in Fig. 1, can be considered as the consent of the user. Moreover, the recording of informed consent is performed as followed: (i) the controller requests for consent through an access policy request; (ii) the Pod matches the request with the stored user policies; (iii) in the case where a match is found, the application UI presents the user with a message such as *'**Application X** wants to access **data Y** for **purpose Z**. This request matches your preferences to allow access to **data Y** for **purpose Z**. Grant or refuse access?'*, and otherwise a similar message will be displayed without the matching user preference; and (iv) the user's choice in consent is stored and used in further access control requests solicited by the application in question. Automatic authorisations can also be performed over the matching of user preferences and apps policies, however this cannot be considered as informed consent. For instance,

taking into consideration the requests in Listing 3 and Listing 4, the application should specify that Listing 4 requires explicit consent from the user since it is directly related to sensitive categories of data, while Listing 3 is related to email and social networks information and therefore automated authorisation can be permitted.

The existing Notification Mechanism should also be updated to allow for update or revoke requests regarding consent and the Reasoner component should be extended to be able to reason over ODRL policies stored on the ACL files.

Fig. 2. Architecture of the proposed solution.



The User Interface, along with the Authentication and Inbox features already provided by the existing Solid specification, proposes the addition of two components: Metadata and Policies Editors. These components will assist users to craft granular policies for the management of access to their Pod, without burdening them with technicalities of writing ODRL policies. The proposed Service client should also implement an Inbox feature for the notification mechanism already discussed and additionally a Policies Editor for the crafting of access permission requests based on the proposed ODRL profile.

VI. Discussion on Challenges

Efficiency and performance: Real-world practices might see the Solid Pod receive and work with a large collection of policies in the form of preferences, requests, and authorisations. This may require limiting features to a subset of the possible ODRL features to ensure optimal performance of the policy conflict resolution and the reasoning processes, as well as to facilitate efficient querying of policies and authorisations stored within the Pod. Additionally, Pods are a decentralised technology with no guarantees about their storage or implementation environments. Therefore, in addition to the efficiency of algorithms, the different components also need to be tested on different Pod clients and server infrastructures to ensure correct functionality. Given that this involves sharing of personal data, simulated use-cases can provide a safe environment for testing.

Complexity of ODRL policies: To ensure the mechanisms we propose work correctly and efficiently, it is necessary to understand the complexity of the authorisation process and to establish guarantees regarding its operations. While the profile that we have defined increases the expressiveness of policies, one can argue that these policies, when used to represent real-world use-cases, might became too complex. Therefore, the more data is transmitted in policies and requests are made, the more time and resource-consuming will be the authorization checking mechanism. In addition, the resolving of authorization checks over policies in other scoped fashions, i.e. global prohibitions outweigh local permissions, might result in a complex reasoning issues. Given these challenges, one solution can be to limit the complexity of the policies to a 'reasonable' limit and restrict the usage of nested policies. Another potential solution is to adopt a 'minimal' known set of concepts within the policy for efficiency of reasoning, as demonstrated by SPECIAL's compliance-checking reasoner.

Storage and Management of Policies: In our approach, we have utilised the existing ACL storage (file) to also host the policies for request and user preferences. This was based on the existing pattern for storing authorisation policies in Solid's implementation. However, this approach has drawbacks in that anyone with access to the ACL file can view or change the data sharing preferences of the user - which can pose a security and privacy risk. Additionally, the discovery of whether a policy exists for a given resource is currently undertaken by iteratively checking for a corresponding ACL file attached to that resource or its parent - which can complicate the policy finding and resolution processes due to complexity and number of iterations. A potential solution for both drawbacks could be to dedicate a separate protected area of the Pod to house a local database/datastore containing the policies and metadata declarations. Another challenge for users can be the crafting of policies, which can be solved with a 'Policies Editor' component that supports users in drafting the policies, without having the need for previous ODRL knowledge, or other UI/UX component that assists users in authorising apps based on policies

stored, e.g., you have expressed preference not to share this information and this app is asking for that information, how should I proceed?

Policies and Personal Data: If the policies allow discovery of all categories of personal data in a Pod by means of automated requests or a listing mechanism, it can result in unintended risks to the user by way of disclosing unintended existence of personal data. This raises the question of how and whether apps should be allowed to query through discovery mechanisms, or to put it in another way - what restrictions should be placed on apps and policies to ensure data (or categories) are not unintentionally 'leaked'.

Legal Implications of Requests: The use of legal vocabulary and concepts within the policies is done with the intention of invoking legal concepts. While the idea is to hold organisations accountable and make data sharing practices transparent, it also opens a can of worms in relation to other legal bases. For example, if the request from an app specifies that it needs some personal data for "legitimate interests", or even more worrisome - for "fulfilling legal obligation", it is debatable whether the Pod should automatically grant access to the resource. In cases where it does not grant access, it is further unclear whether the liability falls to the user or the Pod provider. For centralised datastores, the app has carte blanche i.e. full discretion to utilise this data under those legal bases without prior request or permission from the user. With a lack of legal regulation and precedence in case law, this issue is unlikely to be satisfactorily addressed.

Legal Interpretation of Request and Preferences: It is important for policies, requests, and preferences to be mapped and clarified in terms of their legal interpretation, e.g. if they can indicate consent. Without such clarity, it is unclear what terms and conditions or legal obligations are applicable where a request is made by an app and an user agrees, or where an user expressed their preference. Additionally, derived from the legal basis, additional obligations and rights can be triggered based on jurisdictions - which will need to be factored in when designing the APIs and interfaces based on Solid's Pod-based infrastructures. For example, regarding consent, it may necessitate requiring the individual to explicitly agree (via manual human interaction instead of automation) that they agree to the proposed request. Similarly, the interpretation whether absence of information should be considered opt-in vs opt-out is necessary.

Consent: To ensure that consent is informed and explicit, specific information items should be provided to the users and recorded in the Pod so that the users have access to their consent authorisations. Furthermore, the Pod should have in place methods to allow the users to update or revoke consent actions previously given. Also regarding consent, it can be debated whether the implicit consent from the established user preferences is enough to provide automated access to non-sensitive personal data such as social network account information. These challenges can be tackled by allowing users to choose which data types, and perhaps purposes, they are comfortable with enabling automation and which not, and also by having a queryable Consent datastore which can be easily updated.

VII. Conclusion

The design decision of Solid for access control, ACL, has the benefit of simplicity –ACL is easily understood and the implementation is straightforward. However, this technology may fall short in two situations: (i) when the user wants to express more complex policies beyond a yes/no; (ii) when the actions must be aligned to GDPR concepts. This paper has shown how the joint use of two specifications of the W3C sphere, ODRL and DPV, can help in these two cases.

Big challenges lie ahead: (i) implementing reasoners that can efficiently perform the authorization decision; (ii) defining the RDF SHACL shapes that neatly determine which ODRL expressions can be evaluated; (iii) declaring mappings to other languages that grant interoperability with compliance tools, Data Management Plan support tools such as Argos[16] and (iv) granting seamless operation with non-ODRL Solid Pods.

Acknowledgements

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) and 'Datos 4.0 Retos y Soluciones' (TIN2016-78011-C4-3-R). Harshvardhan J. Pandit is funded by Irish Research Council Government of Ireland Postdoctoral Fellowship Grant#GOIPD/2020/790; European Union's Horizon 2020 research and innovation programme under NGI TRUST Grant#825618 for Project#3.40 Privacy-as-Expected: Consent Gateway; and by the ADAPT SFI Centre for Digital Media Technology which is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant#13/RC/2106_P2.

References

[1] M. C. Suárez-Figueroa, A. Gómez-Pérez, and B. Villazón-Terrazas, "How to write and use the ontology requirements specification document," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2009, pp. 966–982.

[2] M. Fellmann and A. Zasada, "State-of-the-art of Business Process Compliance Approaches: A Survey," *EMISA Forum*, vol. 36, no. 2, pp. 45–48, 2016.

[3] C. M. de Oliveira Rodrigues, F. L. G. de Freitas, E. F. S. Barreiros, R. R. de Azevedo, and A. T. de Almeida Filho, "Legal ontologies over time: A systematic mapping study," *Expert Systems with Applications*, vol. 130, pp. 12–30, 2019.

[16] https://argos.openaire.eu/

[4] S. Kirrane, A. Mileo, and S. Decker, "Access control and the resource description framework: A survey," *Semantic Web*, vol. 8, no. 2, pp. 311–352, 2017.
[5] K. Fatema, E. Hadziselimovic, H. J. Pandit, C. Debruyne, D. Lewis, and D. O'Sullivan, "Compliance through Informed Consent: Semantic Based Consent Permission and Data Management Model," in *PrivOn@ ISWC*, 2017.
[6] L. Piras, M. G. Al-Obeidallah, A. Praitano, A. Tsohou, H. Mouratidis, B. G.-N. Crespo, J. B. Bernard, M. Fiorani, E. Magkos, A. C. Sanz *et al.*, "DEFeND architecture: a privacy by design platform for GDPR compliance," in *International Conference on Trust and Privacy in Digital Business*. Springer, 2019, pp. 78–93.
[7] R. Iannella, M. Steidl, S. Myles, and V. Rodriguez-Doncel, "ODRL Vocabulary & Expression 2.2: W3C Recommendation, 15 February 2018," 2018.
[8] R. Iannella and S. Villata, "ODRL Information Model 2.2," *W3C Recommendation*, 2018.
[9] S. Agarwal, S. Steyskal, F. Antunovic, and S. Kirrane, "Legislative compliance assessment: framework, model and GDPR instantiation," in *Annual Privacy Forum*. Springer, 2018, pp. 131–149.
[10] M. De Vos, S. Kirrane, J. Padget, and K. Satoh, "ODRL Policy Modelling and Compliance Checking," in *Rules and Reasoning*, P. Fodor, M. Montali, D. Calvanese, and D. Roman, Eds. Cham: Springer International Publishing, 2019, pp. 36–51.
[11] S. Kirrane, J. D. Fernández, W. Dullaert, U. Milosevic, A. Polleres, P. Bonatti, R. Wenning, O. Drozd, and P. Raschke, "A Scalable Consent, Transparency and Compliance Architecture," in *Proceedings of the Posters and Demos Track of the Extended Semantic Web Conference (ESWC 2018)*, 2018.
[12] P. Westphal, J. D. Fernandez, and S. Kirrane, "SPIRIT: A Semantic Transparency and Compliance Stack," in *Proceedings of the 14th International Conference on Semantic Systems (SEMANTiCS)*, 2018, p. 4.
[13] G. Havur., M. Sande., and S. Kirrane., "Greater Control and Transparency in Personal Data Processing," in *Proceedings of the 6th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,*, INSTICC. SciTePress, 2020, pp. 655–662.
[14] M. Palmirani, M. Martoni, A. Rossi, C. Bartolini, and L. Robaldo, "PrOnto: Privacy ontology for legal reasoning," in *International Conference on Electronic Government and the Information Systems Perspective*. Springer, 2018, pp. 139–152.
[15] L. Robaldo, C. Bartolini, and G. Lenzini, "The DAPRECO knowledge base: representing the GDPR in LegalRuleML," in *Proceedings of The 12th Language Resources and Evaluation Conference*, 2020, pp. 5688–5697.
[16] G. Lioudakis, D. C. BAK, N. D. SLG, and M. H. TUE, "Project BPR4GDPR Deliverable D3.1: Compliance Ontology," 2019.
[17] E. I. Papagiannakopoulou, M. N. Koukovini, G. V. Lioudakis, N. Dellas, J. Garcia-Alfaro, D. I. Kaklamani, I. S. Venieris, N. Cuppens-Boulahia, and F. Cuppens, "Leveraging ontologies upon a holistic privacy-aware access control model," in *International Symposium on Foundations and Practice of Security*. Springer, 2013, pp. 209–226.
[18] H. J. Pandit, K. Fatema, D. O'Sullivan, and D. Lewis, "GDPRtEXT - GDPR as a Linked Data Resource," in *The Semantic Web - European Semantic Web Conference*, ser. Lecture Notes in Computer Science. Springer, Cham, Jun. 2018, pp. 481–495.
[19] H. J. Pandit, C. Debruyne, D. O'Sullivan, and D. Lewis, "GConsent - A Consent Ontology Based on the GDPR," in *The Semantic Web*, ser. Lecture Notes in Computer Science, P. Hitzler, M. Fernández, K. Janowicz, A. Zaveri, A. J. Gray, V. Lopez, A. Haller, and K. Hammar, Eds. Springer International Publishing, 2019, pp. 270–282.
[20] H. J. Pandit, A. Polleres, B. Bos, R. Brennan, B. Bruegger, F. J. Ekaputra, J. D. Fernández, R. G. Hamed, M. Lizar, E. Schlehahn, S. Steyskal, and R. Wenning, "Creating A Vocabulary for Data Privacy," in *The 18th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE2019)*, Rhodes, Greece, 2019, p. 17.

APPENDIX

TABLE III
DEVELOPED ONLINE MATERIAL.

Prefix	Namespace
Project webpage	https://protect.oeg.fi.upm.es/solid-consent/
Profile documentation	https://w3id.org/oac/
Code repository	https://github.com/besteves4/odrl-access-control-profile