information

Article

Assessing the Solid Protocol in Relation to Security and Privacy Obligations

Christian Esposito [1], Ross Horne[2,*], Livio Robaldo [3], Bart Buelens [4] and Elfi Goesaert [4]

[1] Computer Science Department, University of Salerno, 84084 Fisciano, Italy; esposito@unisa.it
[2] Department of Computer Science, University of Luxembourg, 4365 Esch-sur-Alzette, Luxembourg
[3] Legal Innovation Lab Wales, Hillary Rodham Clinton School of Law, University of Swansea, Swansea SA2 8PP, UK; livio.robaldo@swansea.ac.uk
[4] VITO, 2400 Mol, Belgium; bart.buelens@vito.be (B.B.); elfi.goesaert@vito.be (E.G.)
* Correspondence: ross.horne@uni.lu

Abstract: The Solid specification aims to empower data subjects by giving them direct access control over their data across multiple applications. As governments are manifesting their interest in this framework for citizen empowerment and e-government services, security and privacy represent pivotal issues to be addressed. By analysing the relevant legislation, with an emphasis on GDPR and officially approved documents such as codes of conduct and relevant security ISO standards, we formulate the primary security and privacy requirements for such a framework. The legislation places some obligations on pod providers, much like cloud services. However, what is more interesting is that Solid has the potential to support GDPR compliance of Solid apps and data users that connect, via the protocol, to Solid pods containing personal data. A Solid-based healthcare use case is illustrated where identifying such controllers responsible for apps and data users is essential for the system to be deployed. Furthermore, we survey the current Solid protocol specifications regarding how they cover the highlighted requirements, and draw attention to potential gaps between the specifications and requirements. We also point out the contribution of recent academic work presenting novel approaches to increase the security and privacy degree provided by the Solid project. This paper has a twofold contribution to improve user awareness of how Solid can help protect their data and to present possible future research lines on Solid security and privacy enhancements.

Keywords: distributed knowledge graphs; social linked data; Solid; privacy; security; data protection; authentication protocols

Citation: Esposito, C.; Horne, R.; Robaldo, L.; Buelens, B.; Goesaert, E. Assessing the Solid Protocol in Relation to Security and Privacy Obligations. *Information* **2023**, *14*, 411. https://doi.org/10.3390/info14070411

Academic Editors: Harshvardhan J. Pandit, Rob Brennan and Victor Rodriguez Doncel

Received: 31 May 2023
Revised: 23 June 2023
Accepted: 13 July 2023
Published: 16 July 2023

1. Introduction

The SOcial LInk Data (Solid) protocol [1,2] is a draft specification for managing personal data on the Web. Solid was proposed to decentralise social networking and take data out of the hands of corporations while enhancing *data sovereignty*, i.e., empowering data owners regarding access to their own data by leveraging reusable W3C standards for the Semantic Web. The Solid protocol standardises interfaces between Solid apps that use data, Solid pods that store data, and services that issue identities to users and other agents participating in the Solid ecosystem. In doing so, Solid brings a layer of trust, authentication, and authorisation to the Web, which are intended to ensure the contextual integrity of information flows through the Solid ecosystem.

The ongoing development of the specifications defining the Solid protocol has been led by Tim Berners-Lee and the Solid Community Group at MIT since 2016. At the time of writing, the process of formalising the relevant standard as W3C recommendations has been initiated, making this paper a timely refection of the current status of drafts at a moment when the standards are expected to be tightened while taking into account, in particular, the experience of a network of developers that are already deploying Solid-based solutions. One such Solid-based solution, described in this paper, is a concrete Solid-based

quality-of-care survey on the We Are platform (https://we-are-health.be/en accessed on 12 July 2023) developed by the team of the authors based at VITO for use in conjunction with Belgian hospitals. This use case we will use to illustrate why privacy is paramount in applications handling personal data, thereby justifying the need for the more general privacy analysis that we provide.

This paper investigates the use of Solid in conjunction with in-force legislation, while exemplifying the interrelation between the two with respect to the European General Data Protection Regulation (GDPR), the principal EU regulation for data protection and privacy. In addition, relevant industrial standards will be considered to identify measures and controls to be integrated into the Solid project in order to provide security and privacy. In particular, this paper conducts an initial analysis to assess GDPR legal requirements in Solid, fit to enable legally acceptable uses of the protocol by whoever stores or processes personal data within the borders of the European Union. However, our methodology is general enough to be replicated in other regulations and/or technologies in the future.

While privacy is a stated goal of Solid, being compliant with specific legislation on the topic is unattainable given the current status of the Solid specification. Even if a specification contains some protection means to safeguard users' privacy, compliance with GDPR requires the provision of detailed information about the purpose, legal basis, etc., of personal data processing, to follow specific procedures, e.g., about maintaining a record of processing activities or about notifying personal data breaches to the supervisory authority, and to guarantee rights such as the right to be forgotten, the right of data portability, etc.

Nevertheless, while GDPR provides a well-defined list of all such legal requirements, it does not specify how and to what extent these are implemented in real-world scenarios. This is due to the fact that the scenarios to which the regulation applies are too many in number, and worst of all, most of them were unpredictable at the time in which the regulation was drafted because they depend on IT technologies that evolve over time.

Incompleteness and vagueness as such are indeed found, in different ratios, in every legislative document [3]. Law usually contains plenty of uninformative expressions that must be interpreted in context, which makes it rather difficult to develop AI-based solutions truly useful for the legal profession [4,5]. For example, GDPR specifies that controllers must take "reasonable steps" or "appropriate measures" to meet the legal requirements. However, it does not specify how these steps and measures are *concretely* implemented in context, fit to check compliance with these legal requirements.

For this reason, legislative documents usually specify which appointed authorities are in charge of monitoring the state of the art and define *operational* requirements that implement the legal ones. For instance, GDPR appoints the European Data Protection Board (EDPB) and the Data Protection Authorities (DPA) of the EU Member States to release further guidelines and recommendations of its norms (see, e.g., Art. 70(1)(d)), encourage associations and other bodies representing categories of controllers/processors to prepare codes of conducts (see, e.g., Art. 40), etc.

Other authoritative bodies such as the International Organisation for Standardisation (ISO) or the National Institute of Standards and Technology (NIST) may release further guidance and standard practices that, although not legally binding, enable organisations that adopt them to argue in favour of their proactive attitude and best efforts to be compliant according to the state of the art in a certain domain [6,7].

Ultimately, it is up to judges in courts to decide what is the most applicable legal interpretation of the norms in a certain context, although in the European legal framework, legal interpretations from jurisprudence are not legally binding either, as they might be subsequently overridden by other judges.

The contextual interpretation of GDPR legal requirements into operational requirements calls for a holistic methodology, which takes into account the additional non-legislative documents mentioned above (recommendations from appointed authorities, codes of conduct, jurisprudence, etc.), most of which, as explained above, are not legally

binding, but they can still provide *presumption* of compliance, i.e., proactive attitude and best efforts to safeguard the personal data and their owners' rights, as explained above.

The methodology consists of two steps. The first step creates a table of legally binding requirements related to the technology under examination—in our case, the Solid protocol. This table makes references to certification schemes and codes of conduct, which are texts that offer some specificity in view of the abstract GDPR requirements. In that regard, the documents that have been used as guidance in view of the abstract requirements of GDPR are the EU Cloud Code of Conduct (EU Cloud) [8], the Data Protection Code of Conduct for Cloud Infrastructure Service Providers (CISPE) [9], and the GDPR-CARPA certification scheme [10]. These texts are not legally binding, but they have been approved by the competent DPAs. In view of this, the first step of our research extracts GDPR obligations from these documents that have received authoritative approval.

The second step discusses, for each requirement collected in the first step, the extent to which Solid technically fulfils it. The discussion focuses on the concrete technological choices made in the protocol; their relation with international standards from ISO, NIST, etc.; critical analyses from contemporary literature in the field, etc.; and, of course, the pros and cons of possible mitigation measures to better address GDPR legal requirements.

Such a coverage analysis could also include jurisprudence; however, in the case of Solid, this jurisprudence is not yet available because the protocol is still an emerging technology not yet consolidated in the market.

The key contributions of this paper are as follows:

- The identification of a class of Solid-based systems where the data subject is the owner of a Solid pod, and the app or app user is involved in processing activities. We argue why this is the priority scenario where GDPR applies to Solid.
- A mapping, between the actors in the Solid ecosystem and concepts in GDPR, drawing attention to the different controllers and their obligations in this distributed system. This mapping is used to extract requirements from GDPR that are grounded in officially approved documents.
- A substantial real-world Solid-based case study to which our mapping and requirements apply, demonstrating that our analysis is of a practical nature.
- A detailed technical security and privacy analysis of the requirements above, in relation to the draft specifications that define the Solid ecosystem at the time of writing. Existing measures to address the requirements in the Solid specifications and ISO standards are discussed.
- The above analysis is used to support the case for novel emerging measures proposed by the authors and in related work. Those measures address security and privacy concerns in the specification at the time of writing, and suggest how to enhance Solid as a tool for facilitating GDPR compliance of actors participating in the Solid ecosystem.

Outline. The paper is structured as follows. Section 2 provides an overview of the current technical status of the Solid ecosystem, and explains why the scope of this work is restricted to scenarios where pods are used to store personal data about the owner of the pod. Section 3 maps actors in the Solid ecosystem to roles in GDPR and extracts primary requirements from GDPR and related officially approved documents relevant to Solid, and notably, Section 3.2 describes a real-life case study involving personal health data and Solid in which our mappings are relevant, and indeed required, in order to justify that such systems are GDPR compliant. Section 4 maps the set of requirements of security and privacy measures in the Solid ecosystem and also assesses emerging proposals by the authors and related researchers to improve coverage of requirements. Section 5 draws attention to emerging legislation, yet to be officially approved, that may impact Solid in the future.

2. Background: Overview of Solid Pods as Employed to Enhance Data Sovereignty

This section provides a technical summary of the Solid protocol and its ecosystem. In Solid, data is gathered in an online storage system accessible via an API, called a Solid

pod, or simply a pod. This paper focuses on typical scenarios where the Solid pod is used to store *personal data about the owner of the pod*. There are other scenarios where the owner of the pod may store data about other people, e.g., data collected via CCTV or social networking data regarding others. In the first case, e.g., data collected via CCTV, the pod owner is indeed a *data controller*, so they are subject to GDPR obligations very different from the GDPR rights and obligations they are subject to in cases where they are the data subject. In the second case, e.g., social networking data regarding others, such as posting on Facebook pictures of friends taken during a party at the pod owner's house, Opinion 5/2009 [11] of the Working Party of Article 29, later re-structured and re-named into the European Data Protection Board (EDPB), has established that GDPR does not apply. In cases where the posted content contains particularly sensitive information, other norms could apply, e.g., norms intended to safeguard human dignity, but not GDPR. This paper will not consider these scenarios. As pointed out above, we will only consider here scenarios in which the pod owner is the data subject and the stored data is their personal data only.

Part of the rationale for focusing on scenarios where the Solid pod contains the personal data of the owner is that Solid has been put forward as a solution for enhancing data sovereignty. The basic idea is that companies that store personal data maintain a copy of the data concerning the data subject inside a Solid pod owned by the data subject. This way, data held about data subjects become more transparent, being gathered together in a system where the data subject has direct access control. Solid provides technical means for the owner to grant or revoke access permissions to use the data in the pod. This way, Solid becomes a tool for enabling data sovereignty by serving as an interface between organisations wishing to be transparent about the personal data they hold and process.

For the scenarios we focus on, besides the owners of pods, we are interested in *Solid apps*. Solid apps are used by *agents* to connect to pods using the Solid protocol. Our focus in this work is on scenarios where the owner or user of an app represents an organisation and is engaged in processing activities. To distinguish from other scenarios (such as when the pod owner manages their own pod via an app), we employ the term *data users* for app users involved in processing activities.

2.1. Technical Components of the Solid Protocol

The Solid ecosystem is a distributed system providing services to establish trust relations necessary to deliver personal data. These services are described in an evolving suite of draft specifications (Solid Technical Reports https://Solidproject.org/TR/ accessed on 12 July 2023), from which we extract this overview. Taking care of trust and privacy issues is all the more important for Solid since, traditionally, knowledge graph technology, such as dereferenceable linked data and Resource Description Framework (RDF) databases, were designed with open data in mind where data is published online under suitable licenses. Thus, if we directly reuse existing technology, some of the guidelines and standard configurations are fundamentally opposed to privacy requirements.

Figure 1 highlights some of the most important phases of the Solid protocol.

Phase 1: Authentication of User Agents That Use Solid Apps

Before personal data in a Solid pod can be transmitted, the first step is for some agent (e.g., the data user or pod owner) to log in to a Solid app. This is achieved using an authentication protocol that checks that the Solid app, the agent, and some issuer for the agent all agree that the login is valid. As part of these protocols, the agent, issuer, and app will mutually agree on the content of an *ID token* issued to the app. The ID token asserts cryptographically the Solid app that the agent is logged in to and the scope of the operations the Solid app can perform on behalf of the agent. The issuer (also known as an identity provider) acts as an external authority, e.g., an employer of the agent, a government agency, etc., that vouches for the identity of the agent, either asynchronously by issuing a credential, or interactively by participating in the creation of the ID token, depending on the specific authentication protocol. The Solid app may verify the credentials

of the agent in relation to the scope of the ID token according to the internal policy of the app, concerning which agents may use the app and in what way. Notice that the entities involved (app, agent, and issuer) are all external to the Solid pod, and notably, the policy of the Solid app is internal to the app and not specified in the Solid protocol. The resulting *ID token* is used next in the authorisation phase when accessing resources in the Solid pod.

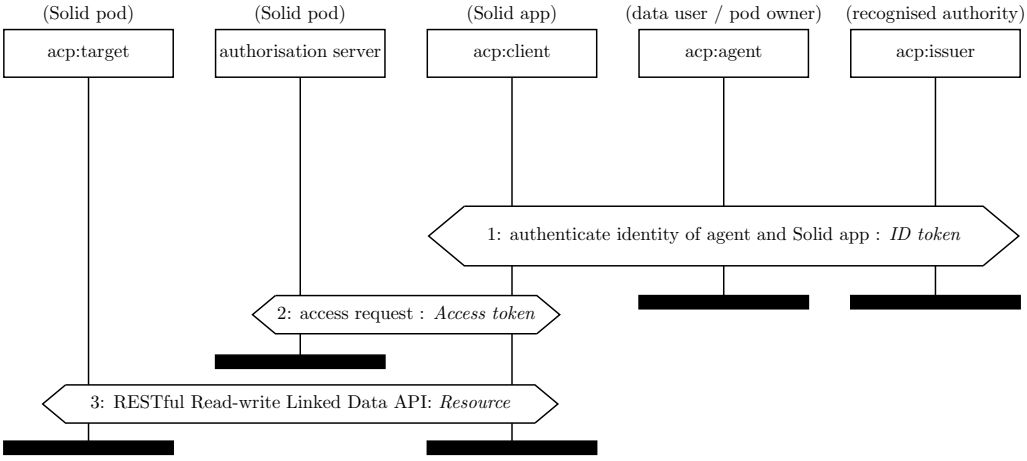

Figure 1. Key phases of the Solid protocol: (1) authentication of (processing) agent by (processing) app; (2) Solid access control; (3) RESTful API for accessing resources.

Several protocols can be used for authentication. The normative protocol mentioned in the draft Solid protocol specification at the time of writing is Solid OpenID Connect [12]. This is a modified flow for OpenID Connect [13], where OpenID Connect is a protocol most people already use regularly, e.g., to log into a website using your social media account as an issuer without entering a password into the website. A feature specific to the Solid OpenID Connect flow is the use of public key cryptography between the Solid app and issuer, rather than a shared secret to establish trust. This allows Solid apps and issuers with no prior trust relationship to cooperate. The public keys required to achieve this are supported by another layer of Public Key Infrastructure (PKI). The current dominant proposal for this PKI is to use WebIDs [14], which are simply HTTPS URIs that serve as a *strong identity* for an entity (data user, pod owner, app, issuer, etc.), which resolves to an RDF document. Note that, in the privacy literature, a strong identity is anything that uniquely identifies an individual amongst a set of individuals. If that RDF document contains a public key, where the agent identified is solely in possession of the corresponding secret key, then WebIDs serve as PKI, under the assumption that the organisation owning the TLS certificates of the domain is trusted to govern the security of the infrastructure supporting the namespace in which the HTTPS URI appears. In addition, the WebID of the Solid app (or the Client ID document) must state explicitly an HTTPS callback URI in a namespace controlled by the app, which is used by the app to receive responses within the Solid OpenID Connect protocol. Since these ingredients are essential for the security of the authentication protocol, such obligations pertaining to WebIDs must be made explicit to those participating in the Solid ecosystem.

Besides Solid OpenID Connect, there are historical and emerging alternative authentication protocols. In earlier WebID-based protocols [15], notably WebID-TLS and WebID-TLS+Delegation, the Solid app would trust the agent based on a cryptographic proof of ownership of the secret key corresponding to a public key presented in the WebID of the agent, without necessarily the presence of an issuing authority [16]. There are also emerging proposals based on Verifiable Credentials (VCs) [17], which, if used appropriately, allow the issuer to vouch for the identity of an agent without being able to trace the uses of the credential to log into an app. Further candidate mechanisms include authentication flows, making use of DIDs [18] resolved via a blockchain rather than HTTPS WebIDs,

and stronger mechanisms, such as USB-based authentication. Multiple flows may be supported, depending on security demands of an application, particularly when logins are for administrative purposes.

Phase 2: Authorisation of Solid App by Solid Pod

After Phase 1, once the Solid app is in possession of an *ID token* asserting cryptographically the app, user, issuer, and scope delegating to the app, the Solid app attempts to access the relevant pod(s). In this second phase, the Solid app contacts an authorisation endpoint, which authenticates the app based on the *ID token*. The decision on whether to grant access to the resources that the app requests access is determined by an access control policy. Solid currently supports two separate access control mechanisms based on the Web Access Control (WAC) ontology [19,20] and on the Access Control Policy (ACP) ontology [21]. WAC, the earlier of the two proposals, provides an Access Control List (ACL) ontology for standardising traditional access control lists [22], which map each resource to the agents that can access the resource and the operations they can perform (`acl:read`, `acl:write`, `acl:append`, `acl:control`). The more recent proposal is ACP [21], which features richer *context graphs* for policies and grants that can record more precisely the combination of actors necessary in order to access a resource using the properties `acp:issuer`, `acp:client` (the Solid app), and `acp:agent` (the entity logged in). The entities in the context graph correspond to the roles in authentication protocols such as Solid OIDC or VC protocols, as suggested in Figure 1. ACP also provides a policy vocabulary for describing *authorisation graphs* that describe policies for accessing resources, and an algorithm for determining whether an *access grant graph* conforms to a policy. A Solid pod is equipped with libraries that enable the owner of the pod, or an agent to whom it grants `acl:control` privileges, to define access control policies.

If access is granted by the *authorisation server* to the Solid app for accessing the pod, then an *access token* is issued to the Solid app that can be used by the app to access the resource in the next step. The form of the access token is left unspecified in the current draft specifications. Generally, the authorisation server and the resource server form the Solid pod and so can agree internally on how access tokens are handled. However, it is conceivable that even the resource server and authorisation servers may be separated, perhaps supporting use cases where one authorisation server manages access to multiple resource servers, or an actor is trusted to control access to part of a pod.

Phase 3: RESTful API for Managing Resources over TLS

Having been successfully authenticated and authorised, the Solid app then uses a RESTful API to access resources. This RESTful API builds on the Linked Data Platform (LDP) [23], which maximises the use of HTTP verbs (GET, PUT, POST, etc.) and HTTP response codes for implementing a read–write linked data interface [24–26]. The protocol should run only over HTTPS. Headers communicate the relevant access token

In a nutshell, Solid provides an online repository to store content of any kind (documents, images, videos, etc.) while providing clear and easily manageable means to grant/revoke access permissions on this content. It does not forbid other interfaces, e.g., SPARQL endpoints, to be provided. Pods may also initiate interactions, via notifications [27,28], that inform a previously authenticated agent, authorised to receive notifications, when a relevant change in resources is triggered.

2.2. Pod Providers and Their Role in the Solid Ecosystem

There are other entities involved in the Solid ecosystem, beyond the actors directly required by the Solid protocol, described in the previous subsection. While it is possible to install and maintain your own Solid server, it is expected that a majority of pod owners will employ *pod providers* that provide the infrastructure for Solid pods, essentially as cloud services, as illustrated in Figure 2. Multiple *pod providers* such as Inrupt.net, use.id, or Solidcommunity.net provide infrastructure for hosting and managing pods, allowing

the owner of a data pod to choose their own pod provider. The fact that Solid apps and Solid pods are interoperable regardless of the pod provider or organisation providing the apps can help avoid data lock-in [29] since suitable apps can be used to migrate data between pod providers. Similarly to cloud storage services, pod providers are responsible for network-level logging and maintenance, such as patching on the fly. There are also *app providers* who, by the decoupled nature of Solid, need not be pod providers.

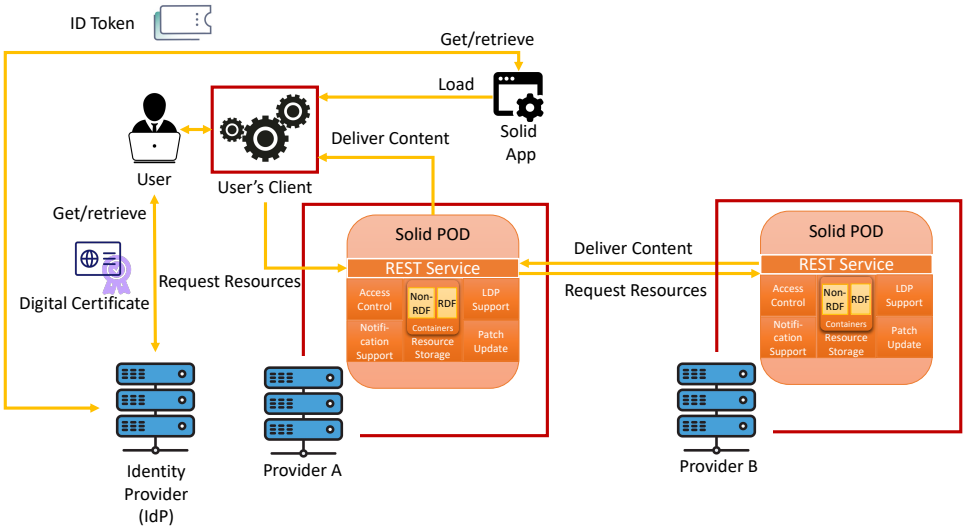

Figure 2. Solid reference architecture.

There are now several deployments of Solid [30]. Notably, there are pilots in the UK and in Flanders that leverage Solid in e-government services to streamline the governmental data processes by putting citizens in control of their own data [31]. Such applications place security and privacy demands on the Solid project, which we scrutinise in subsequent sections. Indeed, simply adopting the proposed Solid protocol, outlined above, is not sufficient to cover all security and privacy requirements and obligations of such government-level data processing. Currently, using Solid to manage personal data does not in itself mean that we have compliance with GDPR; rather, Solid shifts obligations related to the personal data privacy to the various entities within the Solid ecosystem. Without a detailed understanding of legal privacy obligations and technical vulnerabilities across the Solid ecosystem, it is difficult to assess how responsibilities for upholding privacy obligations are distributed amongst the parties involved. We explore next to what extent privacy concerns should not be left to each pod provider and data user, but should be enshrined in the specification of the Solid protocol that *pod providers* and *app providers* implement.

3. Mapping from GDPR to Solid

In the previous section, we emphasised that we are primarily interested in *data users* that connect using *Solid apps* to Solid pods containing data about the owner of the pod. Our contribution in this section is to explain how GDPR relates to such data users, which will allow us to explore the role of Solid as a tool for facilitating GDPR compliance. In scenarios we are primarily interested in, clearly, the pod owner is the data subject in the sense of GDPR. Following GDPR, this entails there are further agents related to the data user, notably a *controller* (typically the management of a company or organisation responsible for the data user) that records the purpose of the processing activities of the data user. Controllers are obliged to respond to complaints regarding processing activities under their mandate. Above the controllers, at the top level, there is a *supervisory authority*, typically at a national or regional level as mandated by EU law, that may issue warnings and fines in the event of non-compliance by controllers. Indeed, the fines

are key to the regulation of GDPR since they incentivise controllers to comply with their obligations, but on the other hand should not be abused to lean on private individuals or small organisations. In 2022, GDPR-related fines were up more than 150% compared to 2021 [32], and in 2023, a record 1.2 billion fine was issued to Facebook alone for GDPR violations (https://edpb.europa.eu/news/news/2023/12-billion-euro-fine-facebook-result-edpb-binding-decision_en accessed 22 May 2023). This section also explains that there will typically be other controllers involved in the ecosystem of Solid, other than the controller responsible for the data user, with quite different obligations.

Next, in Section 3.1, we consider a general mapping from GDPR concepts to entities in Solid in scenarios where the pod holds personal data of the pod owner. However, to emphasise that the mapping is practical, we describe, in Section 3.2, a case study involving personal data, specifically health data, developed by a consortium including VITO. By showing that there is a real-world use case in which concepts in our mapping exist and to which GDPR applies, we emphasise the importance of addressing the requirements, extracted from GDPR in Section 3.3, before such Solid-based systems are released and used to process personal data.

3.1. Identifying the Legal Roles in Solid

GDPR is an EU regulation that applies to EU member states. The regulation applies with regard to the processing of any kind of information relating to identified or identifiable individuals (Article 4(1) GDPR). Solid is a proposal for the management of personal data; thus, obligations deriving from GDPR are relevant. In the present work, we explore these obligations in order to extract certain compliance challenges and propose how the Solid standard can address these challenges. The main premise of GDPR is that it lays down rules for the protection of the data subjects from the threats deriving from personal data processing (GDPR Article 1). In GDPR's terminology, the data subject is the individual whose personal data is processed (Article 4(1) GDPR). GDPR lays down rules that are designed to protect the data subjects. The main addressees of GDPR obligations are the data controllers (Article 4(7) GDPR). Data controllers are first and foremost responsible for compliance with the Regulation [33] (p. 9). This subsection further explains how these roles are allocated within the context of Solid.

3.1.1. Pod Owners

Within the framework of Solid, given that the information stored in the pod can be linked to the pod owners, these are the data subjects. However, we are not taking into consideration cases in which pod owners upload data about other persons, e.g., photos or videos where they appear together with other people or a photo of their friends. We are not considering these cases because, according to GDPR, "the Regulation does not apply to the processing of personal data [...] by a natural person in the course of a purely personal or household activity." (Article 2(1))(c) GDPR). According to this provision, even if a pod owner stores personal data about third persons in their pod, they (most likely) will not qualify as data controllers because processing such data falls under the scope of personal or household activities, in accordance with Article 2(1)(c) GDPR.

This idea is also in line with the Opinion 5/2009 [11] of the Working Party of Article 29, which focus on content, e.g., pictures, posted on social networks. Specifically, in [11], which concerns the role of users in social networking [11], and the conclusion was that "processing of personal data by users in most cases falls within the household exemption" [11] (p. 12). Therefore, according to this view, even if social media users are uploading content about other persons to social media, GDPR hardly ever regards such users as data controllers.

Indeed, as pointed out in Section 2, the pod owners could also be data controllers, e.g., when they store in the pod pictures or videos gathered via CCTV for security purposes; however, we will not consider these cases in this paper. There might be other cases where pod owners become data controllers, for instance, when they start exploiting the information about their relatives in the pod and use them for business purposes, or if

a company directly operates pods as part of their own IoT infrastructure that processes personal data. However, since we anticipate that such cases will be marginal, we restrict our analysis to use cases where pod owners are data subjects.

This perspective is in line with the legal literature on data protection obligations in the domain of decentralised data stores. Specifically, Janssen et al. have convincingly concluded that "[pod users] will, in most cases, be covered by the household exemption for processing done by or with their [pods]" [34] (p. 380). On the basis of these conclusions, we are addressing Solid pod owners as data subjects.

3.1.2. Pod Providers

Having thus identified the pod users as data subjects, it remains to be seen who is responsible for protecting the rights of the data subjects. In that regard, a pod provider is a controller in accordance with GDPR, as they participate in "determining the means of processing" (see Article 4(7) GDPR) by providing their pod services.

It is important that the term "processing" is defined in very broad terms in GDPR. As Article 4(2) mentions, mere storage of personal data qualifies as processing, and it is the pod provider who determines the technical means pertaining to the storage of the pod owner's data. In that regard, there is a need for clarification. Solid claims that it enables data subjects to control their data. However, this verbal expression in the English version of GDPR's text should not lead to the conclusion that the data subjects are the controllers of their own personal data that they store in the pod. This is because, in GDPR terms, a data controller is not necessarily the one who "controls" the data. Instead, identifying who is a data controller, is a question about who is responsible for compliance [33] (p. 9).

Pod providers bear responsibilities that are independent of any access request by a third party. For instance, pod providers are responsible for preventing data leaks from the pod, regardless of whether they have any requests for access to the stored data. Moreover, their action to disclose information by transmitting the data to a third party also qualifies as data processing itself, according to Article 4(2) GDPR.

According to Article 6 GDPR, the pod provider requires a legal basis to process the data of the pod user. In that regard, the pod provider may rely on the necessity for the performance of a contract (Article 6(1)(b)), as there is a contractual relation between them and the pod owner. The present contribution will not further focus on the legal basis on which the pod provider may rely, as this is a matter of legal compliance for the pod provider, and it is not pertinent to the propositions of the present paper.

A clarification is necessary at this point. If the entity that receives the data is located outside of the EU, the pod provider should comply with the requirements of transborder flows of personal data, according to Chapter V of GDPR. Although there are grounds for further discussing Solid specifications in relation to transborder flows of personal data (see, e.g., [35]), the present contribution focuses on cases where both the pod provider and the recipient of the data is located within the EU.

3.1.3. Data Users and Solid App Providers

Within the Solid ecosystem, there are data users who use apps to file requests to pod providers so that they can gain access to the data of the pod owners. If the data users are granted access to the personal data within the pod, then they become data controllers themselves. This is because they decide what kind of processing operations, for example, storage, consultation, use, etc., they will undertake using the personal data. For a list of actions that qualify as personal data processing under GDPR, see Article 4(2).

In this work, we are particularly interested in the controllers that are responsible for apps, and also the controllers that are responsible for the data users that log into apps. The controller responsible for the data users, performing data analytics using data stored in pods, for example, is emphasised in Figure 3. Figure 4 contrasts Solid to a traditional data processing model where there is less transparency concerning processing towards the data subject. We explain each of these two scenarios next.

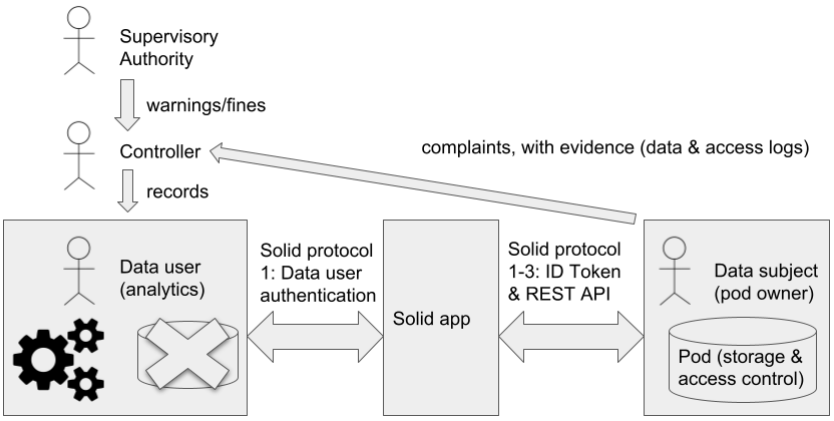

Figure 3. The data sovereignty transition to processing via the Solid protocol. The controller responsible for the data user performing data analytics is emphasised.

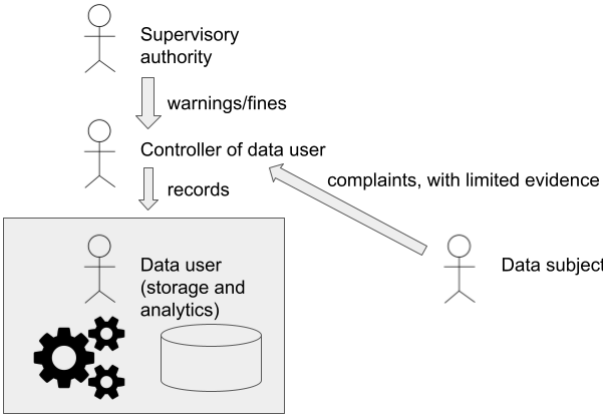

Figure 4. Traditional data processing, where data is stored where it is processed.

Once the data leaves the pod, the entity that receives them processes such data for their own purposes, making them data controllers themselves. In that regard, there are many legal challenges deriving from GDPR, concerning the app users. A comprehensive description of how app users can comply with GDPR fails outside of the scope of the present work, and in any case, proposing in advance how one can comply with the law in all instances is an impossible task. However, by considering the roles of the main actors in the Solid ecosystem, one can further propose solutions for fostering compliance with GDPR.

There might be examples where app users request analytics from the pod owners, as opposed to raw personal data. If the data that they receive is purely aggregate data, then such data is not personal. Despite that, when the app requests the analytics, they are determining some actions to be taken over the data in the Solid pod. For this reason, the app users who are requesting analytics should be regarded as data controllers, even if they are not storing personal data themselves. In a similar case, the European Court of Justice upheld that "defining the criteria in accordance with which the statistics are to be drawn" is sufficient for an entity to fall under the term "controller" in GDPR [36] (Paragraph 36).

Note that while we concluded in the previous subsection that pod providers should be regarded as data controllers, this does not contradict our conclusion that data users and apps are also data controllers, as explained in this subsection. Indeed, a distributed system such as Solid facilitates the separation of responsibilities for different parts of their ecosystem, and therefore, pod providers should not need to act as controllers for all entities interacting with the pods they provide. Of course, an innovative pod provider that also implements apps that gather analytics across pods can also be a controller for an app while being a controller for the storage of the pod itself. However, it can be clean to separate these

two responsibilities, at least conceptually, since one concerns responsibilities regarding storage, and the other concerns other processing activities such as obtaining analytics, which are distinct processing activities with differing responsibilities and hence should be recorded separately.

3.2. Case Study: We Are Health

In order to reinforce our mapping from actors in the Solid ecosystem to roles in GDPR in the previous section, we present a real-world case study using personal health data in deployment of Solid. The organisations involved found it necessary to identify the controllers in the system and their obligations with respect to GDPR in order for the system to go live.

The case study presented here is "We Are" (https://we-are-health.be/en accessed on 12 July 2023), a citizen-centric personal health data platform. "We Are" is based on Solid and operates as a pod provider. An example currently under development is shown in Figure 5 (currently developed as a working demo, to be implemented in a real-world setting in 2024). The case illustrated here is a PREM (Patient Reported Experience Measure) survey to be completed by patients when they leave the hospital after receiving medical care. The PREM questionnaire is developed and maintained by the Patient Organisation, which deploys a PREM app with an electronic questionnaire for patients. Patients completing the questionnaire sign in to the PREM app using their We Are Solid pod, authorising the PREM app to write their answers to their pod. The Patient Organisation does not receive the answers in any way. They are stored in the patients' pods on the We Are platform.

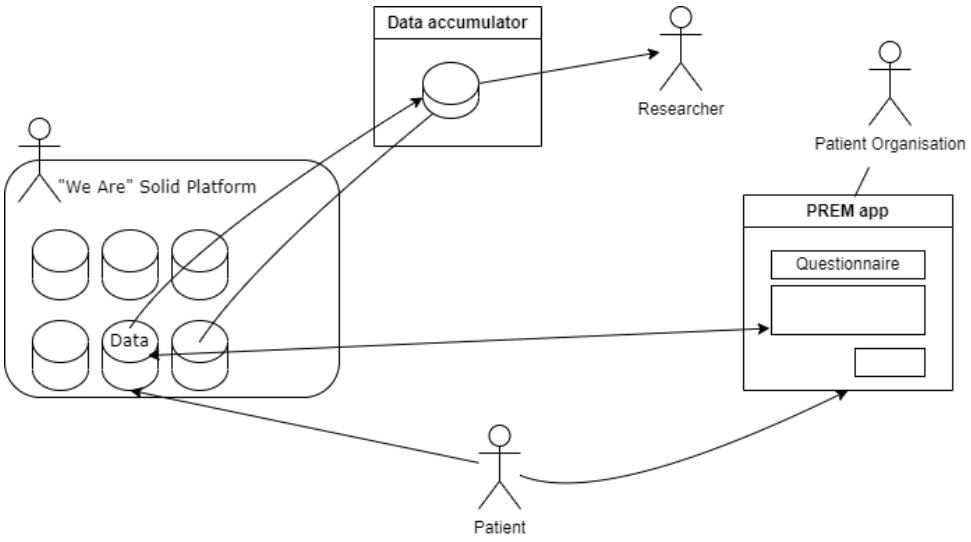

Figure 5. "We Are" architecture. A patient completes the survey using the PREM app, which is developed by the Patient Organisation. Answers to the survey are stored in the patient's pod on the We Are platform. Researchers gain access to this data using a data accumulating app which retrieves data from pods and bundles it into a single file.

The survey answers are used by researchers to produce statistics on quality of care in Belgian hospitals. Patients completing the survey and consenting to making their answers available for this purpose grant access to a data accumulator service that retrieves data from pods and accumulates this in a single file for sharing with the researchers. Researchers do not gain direct access to pods since that would require disclosure of pod URIs, which are identifying. The data accumulation step takes care of pseudonymisation by removing all identifying information, such as WebIDs, other than survey answers.

In terms of GDPR roles, as explained in Section 3.1.3, data users and Solid app providers are data controllers. Likewise, the Patient Organisation providing the PREM app is a data controller since they set the purpose and means of the data collection (i.e.,

setting up the survey). The Patient Organisation does not store the answers to the survey themselves, but these are kept in the patients' pods on the We Are platform.

The platform is not concerned with how the data is collected, and is merely responsible for the safe and correct storage of the survey answers. They therefore act as data processors for the Patient Organisation. However, the We Are platform has its own purpose in its role as a pod provider (see Section 3.1.2), where people can store personal data and give consent to applications, research institutions, and governments to have access to their data. In this capacity, the We Are platform has to process personal information about people with a personal data vault and any personnel they employ. In terms of GDPR compliance, the We Are platform uses strong authentication procedures for access to data pods or the data in the pods, for both pod owners and data users, and encryption of the data. The platform also provides an overview of access grants, with processing purposes, legal basis, and data subject's rights such as revoking access. The researchers' access to the survey data is funnelled through the accumulator app, providing data minimisation and pseudonymisation services.

The researchers seeking access to the survey results to assess quality of care in hospitals are defined in Section 3.1.3 as data users, which makes them again data controllers under GDPR definitions, setting the purpose and means of the data processing. For secure and anonymous processing, they use the services of the data accumulator, who aggregates and pseudonymises the data and makes it available to the researchers. Since the data accumulator only acts out this task for the researchers, it acts as a data processor.

As explained in Section 3.1.1, the control Solid gives over people with data in personal pods. This does not extend to a GDPR-like controller role, and they remain data subjects. In this example, the patients filling in the survey are data subjects, with control over either sharing their responses with third parties or not, but no control over the purpose of the survey itself.

This example illustrates the mixture of data processing roles that were summed up in Section 3.1. As explained in Section 3.1.3, multiple responsibilities come into play in a Solid ecosystem, with data users (the researchers), processing agents (the research organisation employing the researchers and the Patient Organisation), and pod providers (the We Are platform) being controllers on different aspects of the personal data being processed. We also see the complexities at play in the dual role of the pod provider as a controller in one context and processor in another. A service connected to the platform, providing anonymisation/pseudonymisation services to external data users, acts as a processor. In the example, we also highlighted some compliance measures, which we will expand on in Section 3.3, taken in this use case that are not part of the Solid specifications.

3.3. Particular Challenges of Compliance

Having thus identified the roles of data users, app providers, pod providers, and pod owners, we conduct an analysis of the legal challenges of compliance pertaining to Solid specification. Towards that aim, the present work has taken into consideration approved documents in the domain of data protection law such as certification requirements and approved codes of conduct. Given that such documents have been officially approved as containing policies compliant with GDPR, their requirements are a trustworthy source of how DPAs in Europe understand GDPR compliance. The compliance challenges for Solid identified for the purposes of this paper are listed in the following subsections and summarised in Table 1.

Table 1. Security and privacy requirements obtained from GDPR and officially approved documents.

Id	Description	GDPR	Officially Approved Documents
Req_01	Ensure access only to authenticated and authorised users and prevent unauthorised access	Article 5f	II-f-1 GDPR-CARPA; CISPE p. 110
Req_02	Guarantee correct and effective use of cryptography to protect the confidentiality, authenticity and/or integrity of data when stored and/or exchanged	Articles 25(1) and 32(a)	II-f-2 GDPR-CARPA
Req_03	Protect the physical access to infrastructure and protect parameters such as policies.	Articles 25(1) and 32(a)	II-f-2 GDPR-CARPA; see also CISPE p. 59, 109, 111 ; EU Cloud 6.2.7
Req_04	Collection and protection of event logs, by DPO of pod provider, for post-mortem analysis of the system behaviour and user activities.	Articles 25, 24, and 32	CISPE 117; See EU Cloud references to ISO
Req_05	Personal data stored and managed by the system needs to be analysed to determine the purpose for processing	Articles 6, 7, 8, 9, 25, and 35	
Req_06	Actors have different views of access logs. Notably users are allowed to read logs concerning access to their data, and controllers can view logs relevant to their mandate when required. Access logs should be tamper-proof.	Multiple provisions	CISPE p. 25, 63
Req_07	A proper login procedure needs to be put in place at the first use of the solution by having a strong authentication scheme	GDPR Articles 25 and 32	CISPE p. 110
Req_08	In line with data minimisation, unlinkability of accesses and and anonymisation of data should be supported	GDPR Articles 5(c) and 25; see also the definition of pseudonymisation in Article 4(5)	ISO 15408, Relevant, II-e-2 GDPR-CARPA
Req_09	The data subject rights such as the right to erasure, etc., must be respected.	This derives from Articles 16, 17, 18, 21, and 22	On the management of data subject requests in general, see GDPR-CARPA I-8
Req_10	Notification of data breaches	Articles 33–34	EDPB Guidelines 9/2022 on Personal Data breach Notification [37];

3.3.1. Authentication and Access Control

Within the context of Solid, the legal requirements in Table 1 create the obligation of securing access to the stored data within a pod.

First and foremost, it is required that the pod provider ensures access only to authenticated and authorised users and prevents unauthorised access (Req_01). A proper login procedure needs to be put in place at the first use of the solution by having a strong authentication scheme (Req_07). This idea derives from the principles of data integrity and confidentiality, which are laid down in Article 5(1)(f) GDPR, as well as GDPR Article 32, which requires data controllers to implement technical and organisational measures for data security. These provisions have been further specified in the officially approved documents cited in this paper. In view of this, GDPR-CARPA regards proper authentication and access control as necessary requirements for compliance with these particular provisions, and for this reason, it has been expressed as a certification requirement in GDPR-CARPA [10] (Section II-f-1).

In addition, the used authentication should be more robust than a naïve password-based one. For example, CISPE requires that authentication for the use of cloud infrastructure requires user authentication in compliance with a password policy "which has to be aligned with state of-the-art password standards such as minimum complexity, length, password history, lock out in the event of multiple authentication failure or multi-factor authentication" [9] (p. 110). The CISPE code of conduct has been drafted in relation to

cloud infrastructure in general. Despite the differences between cloud infrastructure services and a Solid pod, the conclusions about data security that derive from this code of conduct are pertinent in our case. For meeting these requirements, we further propose Solid specifications about configuration parameters, including identities, security claims, and authorisation policies (Req_03).

3.3.2. Protection of Stored Data

GDPR imposes a duty to implement compliance policies in view of the data stored by a pod provider (see Articles 25(1), 32(a) GDPR). Specifically, data controllers according to GDPR are under the duty to implement technical and organisational measures in order to implement the general principles laid down by the Regulation. In view of this, encryption can be regarded as a technical measure that reduces the risk of unauthorised access ([10] (Section II-f-2) and Req_02).

Apart from technical requirements, GDPR Articles 25(1) and 32 also require the adoption of organisational measures that aim to reduce risks of accidental and unlawful data losses, as well as unauthorised access. These organisational measures go beyond software implementation. In view of this, an important element of compliance is the protection of the physical infrastructure where the data is stored Req_03 [10] (Section II-f-2 [8], Section 6.2.7). In particular, the CISPE code of conduct for cloud services gives concrete examples of such organisational measures, which include the zoning of critical areas and physical authentication of those who access data centres [9] (p. 59).

In conclusion, pod providers should implement both technical and organisational measures for the security of the data of the pod users, with the aim of achieving compliance with GDPR obligations. Encryption and protection of the physical infrastructure are very important compliance requirements in view of these obligations.

3.3.3. Logs for Accountability

Controllers in the system have the duty to keep track of logging activities. This includes controllers responsible for the pod providers, Solid apps, and data users, each with their own mandate. This method has already been recognised as a compliance measure in the domain of cloud computing [9] (p. 117), and we propose this requirement to also apply in the ecosystem of Solid (Req_04 and Req_06). This best practice is an important measure that assists the relevant controllers to comply with the principle of accountability (see GDPR Articles 5(2), 24, 25). Concerning pod providers, by recording the logging activities, the pod provider can prove the means for data subjects owning a pod to check that only authorised users are getting access to the pod. Concerning Solid apps and data users, logs recorded can be used to prove what information they have accessed and in what context, in order to justify their own processing activities.

Event logs concerning the overall functioning of the system, such as network logs and logs recording the server status, can also be retained by the pod provider to adhere to the principle of accountability. Such system records are useful for post-mortem analysis of the system behaviour and some information about user activities. Hence, recording of logging activities can also be a tool to foster the security of the data stored within the pod (see GDPR Article 32).

Moreover, GDPR lays down as a general principle the requirement that the processing of personal data takes place in a way that it is transparent for the data subjects (Article 5(1)(a) GDPR). In addition, the data subjects have the right to obtain from the controller information about the way their data is processed (right of access, GDPR Article 15). In view of these provisions, pod users should have the ability to see their logging data as well as the access requests and the recipients of their data. This best practice has already been established in the domain of cloud services, where clients of cloud infrastructure services have the right to request information about logging activities in their accounts as well as the monitoring of the interfaces with their stored data [9] (p. 117). On the one hand, there is a duty to retain metadata about logging activities, but, on the other hand, such

log data constitutes personal data, pertaining to the pod users. For this reason, log data is itself protected under data protection law, and for this reason, it should be subject to very strict restrictions as to who should be able to access it. For this reason, we propose that only the DPO of the pod provider is entitled to completely read all the logs, which need to be tamper-proof. Moreover, a pseudonym scheme should be implemented so that users are not traceable throughout the logs, and logs are stored on a tamper-proof DB (Article 25(1) GDPR).

3.3.4. Notifications

GDPR lays down some very specific requirements about notifications when data breaches occur. In that regard, GDPR Article 33 requires that data controllers notify the competent data protection authority, and Article 34 obliges data controllers to notify the data subjects about data protection breaches. In view of these provisions, pod providers have the duty to notify the data subjects in case of data breaches involving the pod Req_09. In that regard, GDPR Articles 33 and 34 GDPR lay down details about the content of these notifications. In addition, the EDPB has provided further guidance about data breach notifications [37].

3.3.5. Facilitating Data Subject Rights

According to GDPR, the data subjects have the right to erase their personal data (Article 17). As far as the pod provider is concerned, the Solid protocol is a use case that enables the data subjects to exercise their rights. In particular, data subjects may be able to delete their data from their pod at will.

However, data users who have accessed the personal data within a pod or via an app recall they are data controllers themselves and, furthermore, likely store a local copy of the personal data. Therefore, data subjects have the right to request from such data users to have their personal data deleted locally also by the data user, not only in the pod. It should be mentioned at this point that the right to erasure is not an absolute one. As GDPR Article 17(3) prescribes, there are legal grounds for the data controller to refuse the deletion of the personal data [38] (pp. 11–16). One cannot declare in advance how the data subject deletion requests should be handled by app users, in all possible scenarios. However, what is important in view of the proposals of the present paper is that there should be a mechanism that facilitates such requests. In view of this, we further explore how the app users who accessed certain data can receive notifications about the request of the data subject to have data deleted (Req_09).

The same logic can be deployed in view of exercising rights that GDPR lays down. For example, if the data subject wants to restrict the processing of their data by an app that accessed them (GDPR Article 18), then there is a need for a specification, according to which the app users can receive requests from the pod owner. The same can be said about the data subject's right to object to specific ways of processing their personal data, such as, for example, profiling or automatic decision making (GDPR Article 21). In view of this right, there is the need for a specification that will enable pod owners to convey these requests to the app that accessed their data. Once again, the right to restrict processing or to object are non-absolute rights, which means that it is not always the case that the app users have to uphold the data subject's requests. However, a specification about Req_09 could enable the data subjects to convey their requests about their rights. This would be a valuable safeguard for fostering their rights.

3.3.6. Information That Must Be Provided

GDPR Article 13 lays down the information that should be provided to the data subjects when a data controller obtains their data from them. Within the context of the Solid ecosystem, the data user should provide the information prescribed in this article to the pod owner. Specifically, Article 13 requires that the data subject receives the following information:

(a) The identity and contact details of the data user, as well as the contact details of their DPO (if the data user has one).
(b) The purposes for which applications are accessing data in the pod.
(c) The legal basis according to which applications are accessing the data in the pod.
(d) If the app users are relying on legitimate interest as a legal basis for accessing the data in the pod, then there is a need for recording what is the legitimate interest pursued by the app user.
(e) The recipients of the categories of data, if any.
(f) If applicable the fact that the controller intends to transfer the data to third countries.
(g) The period for which the data will be stored or the criteria for the evaluation of that period.
(h) The existence of the right to request from the controller access to and rectification or erasure of personal data or restriction of processing concerning the data subject or to object, as well as the right to data portability.
(i) If the processing is based on the legal bases of consent, then it is necessary that the pod owners should be informed about their right to withdraw consent.
(j) The right to lodge a complaint with a supervisory authority.
(k) Whether the provision of personal data is a statutory contractual requirement or a requirement necessary to enter into a contract, as well as whether the data subject is obliged to the personal data and the possible consequences of failure to provide such data.
(l) The existence of automated decision making, including profiling, as well as information about the logic behind it and the potential consequences.

In other words, GDPR prescribes that the data users have to provide to the pod owners the information presented in Elements (a)–(l) mentioned above. There are multiple ways that this information can be provided. For example, it can be the interface of the application that provides this information, and the application subsequently requests the data from Solid. However, we will focus primarily on the purposes of processing, the legal bases, and the legitimate interest of the processing of data by the app users (Elements (b)–(d) listed above). Thus, we propose a requirement that addresses the need of informing the data subjects about the purposes, legal bases, and legitimate interests pertaining to the processing (Req_05). Elements (b)–(d) in particular do not exhaust the app user's obligation to inform the data subject, but our proposed specification will ensure that at least some of the necessary information is provided.

Moreover, Solid could be a valuable tool for data users to inform their data subjects about the purposes, legal bases, and legitimate interests pertaining to data processing. In this sense, Req_05 is a valuable tool that assists data users to comply with GDPR. In relation to that, data users have to comply with the principle of accountability which requires that they are in a position to prove their compliance with the law (GDPR Article 5(2)). In view of this obligation, a specification that meets Req_05 could assist data users keep track of all the data that they have accessed as well as the purposes, the legal bases and the legitimate interests, pertaining to all these access requests.

As a conclusion, a specification that addresses Req_05 could serve multiple purposes. First, it fosters the data subject rights, and it enables pod owners to see which of their data was accessed, for what bases and legal bases. Moreover, it ensures that app users are meeting (some) of their GDPR obligations regarding information made available to data subjects. Last but not least, such a specification is a valuable tool that enables data users to keep track of the purposes and legal bases, according to which they accessed personal data for purposes of GDPR compliance.

3.3.7. Data Minimisation and Unlinkability

While some parties must be able to identify or link agents for functionality reasons, in line with the GDPR principle of data minimisation (GDPR Article 5(1)(c)), the parties that can do so should be minimised. For instance, if there is technology available to make

accesses unlinkable from the perspective of the issuer, then data minimisation would suggest that technology should be enabled. This derives from GDPR Article 25(1), which requires the adoption of measures in line with the principles of data minimisation, as described above.

Measures appropriate to adopt for data minimisation purposes are covered by professional standards such as ISO 15408 standard (Common Criteria aka. CC) [39,40]. The ISO 15408 standard is considered within our analysis as it specifies functional and assurance requirements for computer systems. Such standards address security and privacy issues at the root cause rather than treating the symptoms, for example, to encourage the adherence of technology to technical privacy properties. The privacy property *unobservability* in CC can be realised if a user interaction with the system happens without others, especially third parties, being aware that a specific data instance or operation has been used by a given identifiable entity. Solid makes unobservability challenging since some communications, which might, in traditional information systems, be internal to an organisation are exposed over the Web. Another property defined by CC is *pseudo-anonymity*, where users interact with the system without disclosing their identity, but can still be accountable later if obliged by law to trace past activities in the case of a dispute. Pseudo-anonymity is explicitly mentioned also in GDPR as an explicit measure for mitigating the risk of leaking personal data during processing. A highly relevant privacy property, stronger than anonymity, is *unlinkability*, where users may make multiple uses of resources or services over a time frame, without others being able to link these uses together.

4. Security and Privacy Assessment of the Solid Protocol

We now consider at a technical level the requirements extracted from GDPR and other officially approved documents in the previous section by explaining how they are reflected in the Solid protocol. Privacy and data protection are closely related, although they do not refer to the same notion. Specifically, under the interpretation that privacy is related to preserving information in its intended context, free from interference or intrusion from outside that context [41], data protection impacts privacy since it concerns the governance of the context in which data may be used. Security is also not synonymous with privacy, despite considerable overlap. Security may be characterised as the well-informed balance between multiple risks and controls [42].

Authoritative bodies such as the International Organisation for Standardisation (ISO) or the National Institute of Standards and Technology (NIST) release guidance and standard practices for security and privacy that, although not legally binding, enable organisations that adopt them to argue in favour of their proactive attitude and best efforts to be compliant according to the state of the art in a certain domain. Most requirements in Table 1 are also reinforced by security standard ISO 27001:2017 [43], except perhaps those specific to personal data, the legal basis and purpose of processing (Req_05), and notification of data breaches (Req_10). Indeed, ISO 27701[44] (an extension to ISO 27001 and ISO 27002 for privacy information management) presents, in Annex D, a mapping between the introduced ISO controls and those in GDPR. The ISO 27001 standard clearly supports GDPR Article 32 as ISO 27001 defines best practices for mitigating risks within the organisation, while Article 32 indicates that such risks must be taken into account.

The contribution of this section is to present, in Table 2, a systematic assessment of the degree of satisfaction of security and privacy requirements from Table 1 with respect to the current Solid specification and the implementations. We explain to what extent requirements are covered by the Solid specification, and where there are vulnerabilities or potential weaknesses in the system. We identify gaps in the current specifications that may be addressed constructively by evolving the Solid specifications. Some evolutions are under development in related work, while others are proposals of this work. Compromises may be required when requirements are in conflict, e.g., while security demands logs, the data subject has the right to be forgotten.

Table 2. This maps the requirements to measures enshrined in the state of the art and ongoing evolution of the Solid protocol.

Id	Description (Short)	Existing Measures	Emerging Measures
Req_01	Authentication and authorisation	Solid OIDC, WAC, ACP	Must avoid: IdP mixup in Solid OIDC, App mixup in WAC/ACP and Solid OIDC.
Req_02	Effective cryptography	HTTPS in Solid protocol, also signatures in ID tokens in Solid OIDC, etc	Explicitly forbid HTTP. HTTP error codes allow pods to be profiled. Use 401 instead of 404 everywhere. Make cryptography for message exchanges explicit in Solid OIDC.
Req_03	Protect physical access and administrative parameters	ISO 27001 A11. ISO 27002 Sec. 7.	Reference cloud storage standards to make pod providers responsible for infrastructure security. Certify pod providers with respect to security audit status.
Req_04	Event logs for system behaviour and user activities.	ISO 27001, A12	Separate logging of network and server activity for pod providers. A pseudonym scheme can be implemented when logs are aggregated for CISOs/DPOs.
Req_05	Purpose and legal basis	None	Enhance ACP with explicit purpose (ODRL/DPV), following related work. Enhance Solid OIDC to authenticate purpose in scope and record in ID token. Ensure match between authentication and access control. Consider supporting delegation in authentication protocol, e.g., from controller to processor.
Req_06	Tamper-proof access logs with different views	Nothing explicit	Trust: Log trusted evidence, such as signatures on ID tokens, or use trusted environments. Reuse ACP authorisation and grant graphs to policies separately from access grants and access instances. Trust in fine-grained access information via suitably specifying access tokens. Logs made tamper-proof via cryptographic means or by using a trusted 3rd party. Redesign authentication with logging evidence in mind, leveraging VCs as evidence for example.
Req_07	Multi-factor authentication	None	Administrative functions, including first access and account recovery, must use more robust agent authentication than a password.
Req_08	Data minimisation via unlinkability and anonymity	Solid OIDC allows agent to manage multiple WebIDs permitting different degrees of anonymity	Issuer can track and profile agents in Solid OIDC. Unlinkability can be introduced via VC-based protocols. Solid apps can apply (pseudo)-anonymisation.
Req_09	Protect rights of data subject	Data portability (Article 20) addressed when pods used to log personal data	Protocol for communicating logs to controller to enact rights to object (Articles 21 and 22). Protocol required for erasure since data user may retain copy.
Req_10	Notification of data breaches	Linked data notifications alert to changes of resources only. ISO 27001, A16 addresses incident management	Pod owner can generate report that can be trusted in court. Monitor logs and linked data notification for anomalies via app.

4.1. Ensure Access to Authorised Users Only (Req_01)

The normative authentication protocol for Solid mentioned in the Solid protocol specification at the time of writing is Solid OpenID Connect (Solid OIDC) [12,13]. Adopting a flow of Open ID Connect for authentication alleviates some security challenges for Solid apps since they need not handle their own ad-hoc login logic nor store the passwords of users. It also enhances security for users since users need not hold separate identities and passwords across multiple sites, reducing associated risks [45]. The key feature that Solid OIDC brings to OpenID Connect is the use of public key cryptography

between the app and issuer. This allows Solid apps with no previous trust relationship with an issuer (and hence no shared symmetric secret such as a password) to make use of that issuer. This is partly enabled by PKI that maps HTTPS URIs, called WebIDs, to WebID documents containing the public keys of the actors involved, including the app itself. The app must also advertise secure callback endpoints in their WebID documents; otherwise, authentication can be hijacked via a man-in-the-middle attack, where an attacker masquerades as an app of their choice and provides their own malicious callback URI to intercept secrets. These constraints are specified in the Solid OIDC specification and primer [12,46].

Authentication vulnerabilities in Solid OIDC

OpenID Connect is widely deployed with robust libraries, but there remain vulnerabilities that may be addressed in the specification. We review some vulnerabilities that apply to Solid OIDC, below, and explain how they may be addressed by tightening the specification of Solid OIDC.

Issuer Mix-Up. Some flows of OpenID Connect are known to be vulnerable to issuer mixup attacks [47,48], and Solid OIDC is no exception. Recall that in Solid OIDC multiple issuers may be used, some of which may be previously unknown to the app. Even if the relevant Solid app is honest, an attacker may pose as a fake issuer. The attacker can then pretend to be the app in relation to an honest issuer, IdP (Issuer) in Figure 6, which some legitimate user of the app uses to log in, using the password registered by the app in its WebID document, that the honest issuer checks. The issuer then uses a secure callback URI, pre-registered by the honest app, to send a pseudo-randomised *code* to the honest app (along with the $userID$ identifying the user that logged in, but *without* an $issuerID$ as emphasised by it being struck out in Figure 6). At this point, the attacker has not yet intercepted the code; however, since the app was confused and originally believed it was talking to the attacker's fake issuer, the app then tries to exchange the code with the attacker instead of the honest issuer. When exchanging the code, a cryptographic secret session key is also generated by the app, sk_A in Figure 6, which the app is supposed to be able to use later to prove that it has possession of what is received in exchange for the code. However, since the attacker has intercepted the code, it can instead make up its own secret session key, sk_E in Figure 6. This enables the attacker to prove ownership of an ID token which was intended for the honest app but is in possession of the attacker, at the next authorisation step of Solid OIDC involving the authorisation server of some Solid pod; that is, the attacker can now log in to a Solid pod as if it were the honest app with the honest user logged into it.

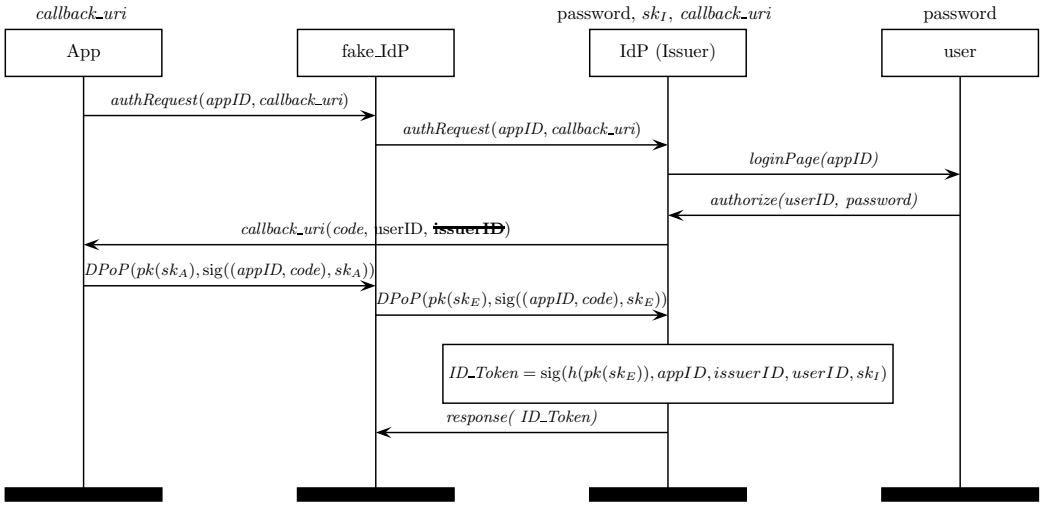

Figure 6. Issuer mixup attack vector enabling an attacker to prove possession of an ID token intended for another app.

This attack can be mitigated by the issuer recording its own identity when the client is redirected back to the Solid app. A standard way of implementing this measure is to add an `iss` field to the HTTP header of the response, as reported in RFC 9207 [49]. To see this, observe that if we introduce the **issuerID** in Figure 6, as suggested by restoring ~~**issuerID**~~ struck out, the app will block the attack since it will be able to spot that the fake issuer is not the issuer that responded. Such clarifications are not yet made in the draft specifications of Solid OIDC.

Further clarifications. We mention two further vulnerabilities known to be relevant to OIDC in general [48]. To protect the credentials of agents from malicious Solid apps attempting to steal passwords, having received credentials from a user, the issuer must never use an HTTP 307 "TEMPORARY REDIRECT" status code since it replays the credentials in the body of the POST to the Sold app. Instead, the issuer MUST use a HTTP 302 "FOUND" or 303 "SEE OTHER" status code having handled credentials from a user. Indeed, the Solid OIDC draft suggest 302, since the HTTP semantics of 303 can be interpreted by some clients as a permanent change of the location of the issuer rather than a redirect as parts of the flow of the protocol. Also, Solid should protect against session hijacking since URIs with external untrusted domains may appear in resources obtained from pods. Solid pods should implement a *referrer policy* that instructs the browser to strip away the state information from the referrer field of the header when accessing such external URIs.

Priority security problem: access control & Solid OIDC.

We observe here a security problem concerning the interaction between the authentication protocols such as Solid OIDC and access control mechanisms, present in the current Solid protocol specification at the time of writing. We believe the relationship between actors in the authentication protocols (currently Solid OIDC) and the access control specifications WAC and ACP should be made more explicit to avoid data breaches. In particular, the owner of a Solid pod, defining a policy, may not wish any Solid app with which an agent authenticates to access some resource intended for the agent in some other context. This means that naming only the agent in the access control policy is insufficient, which is currently the case in ACL, as used by WAC. Instead, ACP must be employed whenever an app is used to access a pod via Solid OIDC. Furthermore, the context graph in ACP, referred to by an authorisation graph that describes access control policies, must explicitly indicate that access is granted to a particular Solid app by using the property `acp:client`. This is not clearly stated in the specification of ACP or Solid. Indeed, the first example provided in the ACP specification at the time of writing indicates only an `acp:agent`, making it insecure for use with Solid OIDC, as we explain next.

Failing to address the above-mentioned issue enables the following attack vector.

1. An honest agent, say *agent1*, logs into an honest app, say *app1*, that is granted read access to a resource, say *resource1*, pod.
2. The authorisation graph indicates `acl:agent` *agent1*, in the case of WAC, or provides a context with `acp:agent` *agent1* and does not provide reference to *app1*.
3. The same user authenticates with any other app, say *app2*, that was not intended to receive *resource1*. That app may be compromised since not all apps may be trusted to be as secure as *app1*.
4. Since the compromised app is authenticated with an ID token referencing *agent1*, and since `acp:client` is absent from the policy of the given resource, the authorisation server protecting resource *resource1* will grant read access to the resource when *app2* requests access.
5. The access token issued by the authorisation server to *app2* will be valid for *app2* to use to retrieve *resource1*, resulting in a data breach.

One cannot trust all apps to be perfectly secure. Thus, a security failure in one app that an agent uses cannot result in all the data intended for the agent across all pods and apps becoming compromised. Thus, the above attack vector must be addressed to avoid data breaches.

The draft concerning Solid Application Interoperability [50] goes some way to address the above by indicating that the subject of an access grant may be an agent and app, indicated as follows.

> An *authorization subject* [snip] is either an Agent [snip], a User-Piloted Application in use by any Agent, or a combination of a specific Agent using a specific User-Piloted Application.

The mechanism could be clarified since there is no explanation of how a combination of both should be indicated (via ACP for example). To address our observations above, we suggest such statements should be strengthened to ensure that a grant towards an agent must be tied to a specific app. There is a community discussion related to this: https://github.com/Solid/web-access-control-spec/issues/81 accessed 13 December 2022.

Going further, the issuer must also be indicated explicitly, to avoid accepting ID tokens where an honest issuer is bypassed entirely, and a malicious issuer manufactures their own ID token for any combination of honest app and user. Even if an authorisation server attempts to mitigate such an attack by looking up the WebID of the agent, and checking the list of permitted issuers, there is no guarantee that there is one issuer, nor that all issuers are equally secure. This would mean neglecting to mention the issuers means that the security of the agent becomes only as good as the security of the least secure issuer listed by the agent, and is not in the hands of the pod, which may trust one issuer more than another. This is a legitimate concern for Solid OIDC, which permits anyone to become an issuer.

A complementary measure is for the agent using the app to explicitly restrict the scope of an ID token granted to an app as part of the authentication protocol. As it stands, if the specification of the Solid OIDC protocol were to be implemented literally, without further refinements, the scope is limited to a single string ‘‘`Solid`’’ [12]. Thus, although the identities of the app, the agent, and the issuer are simultaneously authenticated by both the agent and the app, the app is implicitly delegated by the agent to use the resulting *ID token* to access in any way any resource owned by the agent in any Solid pod. A potential measure is to also include, instead of just ‘‘`Solid`’’, a representation of a policy (e.g., an authorisation graph), indicating an explicitly narrower scope, which means an agent has delegated access to a Solid app. The scope should be able to narrow down the set of pods and resources within those pods that the Solid app has to be approved to access on behalf of the user. To implement this, the authorisation server should only grant an operation for a resource if the policy indicated in the ID token also grants permission. This can help avert data breaches, and hence is pertinent with respect to GDPR from the perspective of any data user logging into a Solid app, who should be able to set appropriate policies themselves to avoid such breaches, and not depend exclusively on the owner of pods.

4.2. Effective Use of Cryptography (Req_02)

In the Solid specification, cryptography is mandated via the use of HTTPS in a RESTful API for transmitting private data between data pods and authenticated and authorised clients and as part of the authentication protocol (Req_01). The use of HTTPS, in a RESTful API for sharing private data between data pods and authenticated/authorised clients, does not necessarily guarantee privacy; hence, we examine how effectively HTTPS is employed in the current Solid specifications. We focus here on which aspects of the RESTful API of Solid should be addressed due to privacy issues.

Regarding effective cryptography beyond HTTPS, GDPR stipulates cryptography can be used to improve trust in the access logs, which is not currently covered by the Solid protocol (see also Req_04). Moreover, Solid does not mandate the use of cryptography when storing data in pods (see also Req_03).

Profiling attacks exploiting REST

We illustrate profiling vulnerabilities here using a leading example scenario. Consider a scenario where a data subject makes resources (e.g., a health record) available to an agent (e.g., a doctor). In addition, to avoid profiling, the data subject would prefer that not even

the existence of the resource should be revealed to third parties. Suppose that the resource in the pod is made available via https://john.provider.net/vaccinationdata.ttl. There is then a crude but effective profiling attack, impacting privacy, where the attacker poses as an app trying to access a resource and observes from the HTTP response whether a resource exists. Suppose that 404 "NOT FOUND" is the response for resources that do not exist, and 401 "UNAUTHORIZED" is used when the app has not yet been authenticated to use a resource. For pods implemented in this way, an attacker can determine that a data subject has a vaccination record even though the attacker has no access to the record, thereby violating privacy. Note that, at the time of writing, the Solid protocol specification indeed states, "when a POST method request targets a resource without an existing representation, the server MUST respond with the 404 status code".

A resolution to the privacy problem identified above is for pods to respond with 401 "UNAUTHORIZED" whether or not the resource exists. That is, a pod should never respond with a 404 "NOT FOUND" even if a resource does not have an existing representation, and should instead respond to such a request with a dummy 401 "UNAUTHORIZED" with a dummy token in its header that is indistinguishable from a genuine 401. This way, pods remain compatible with the current flow of Solid OIDC that makes use of the handle provided in a 401 "UNAUTHORIZED" token to indicate which resource needs to be accessed. This resolution would be important to make explicit for functionality reasons so that apps take into account that 401 could also mean that resource does not exist. The existence of a resource can only be known by prior knowledge, e.g., by the data subject informing the data user, or by authenticating with valid credentials according to the policy of the resource. Non-existence cannot be determined, and not even by attempting to authorise using incorrect credentials and observing the response.

Another issue is that the specification permits HTTP URIs. HTTP URIs are redirected to their HTTPS counterparts using a 301 "MOVED PERMANENTLY" status code and a location header. The specification states that "a data pod SHOULD use TLS connections through the HTTPS URI scheme to secure the communication between clients and servers." However, "SHOULD" should be upgraded to "MUST" since the protocol for handling HTTP URIs reveals the URI that is being requested by honest clients to any eavesdropper. An eavesdropper may further exploit the HTTP URI by injecting their own malicious payload in place of the 301 returned by the honest server. Therefore, the above line should be erased from the specification, or, better still, all HTTP URIs must return a uniform error, such as 501 "NOT IMPLEMENTED". To counter the argument that such status codes may impede users who habitually enter `http:` into their browser, note that it is Solid apps that access pods, and not naïve end users, and hence Solid pods can be expected to implement appropriate API usage.

Even with the above issues addressed, HTTPS traffic between pods and apps can reveal significant information even with effective cryptography in place. This is because an external attacker may nonetheless infer information about the resources accessed. An attacker can trivially infer the fact that `john.provider.net` is contacted by a particular app, hence the domain should not reveal the name of the data subject as a sub-domain (as is the case for Inrupt for example). To reinforce this, also consider that a man in the middle knows the following: the length of the URL being accessed, rounded up to the nearest block length in the cipher, say 128 bits; the length of the resource in the response; and the response time between initiation of the session and termination of the session, from which behaviours may be inferred. Indeed, in reasonably busy applications, such as those with a dynamic Ajax API, such information has been shown to reveal fine-grained information, such as keywords being typed [51]. Even coarse-grained information inferred in this way can be used to trace behaviours profiling a data subject. For instance, if a data subject is a candidate in an election, activity concerning their data may reveal information about their popularity without consent. Avoiding such profiling attacks would require radical steps that perhaps obfuscate the above information. We expect, for now, such measures are out of the scope of the state of the art and resources available to pod providers. The Solid

protocol specification at the time of writing mentions protecting against timing attacks, without being specific.

4.3. Protect Physical Access and Administrative Parameters (Req_03)

Physical protection is the concern of the management policy of *pod providers*, hence largely out of the scope of the Solid-related specifications. One could, however, argue that to promote trust in pod providers, it would be advisable for specifications in the Solid ecosystem to mandate mechanisms protecting against insider threats. The storage of servers should be encrypted, reducing risks associated with attackers with physical access to servers.

There should be a transparent policy regarding who has access to the encryption keys to disks, and under what conditions. Additionally, the data stored in different countries comply with different data protection laws and are required to apply different physical protection mechanisms. Trust in pod providers can also be established by being transparent about the status of the provider with respect to security audits, which could be an important aspect of the governance of the Solid ecosystem. Provisions should be made for transferring pods between providers securely. This requires physical and network security and standardisation of protocols for handover. The current specification has not covered the inter-provider exchange of pods. The pod providers should also ensure that client details and access control policies are in a secured database.

4.4. Event Logs for Post-Mortem Analysis (Req_04)

A *pod provider* does not necessarily have access to the data inside the pod. They provide the infrastructure in which the data is stored and typically employ a *Chief Information Security Officer* (CISO) to oversee the configuration and maintenance of the machines running the servers containing the pods of the provider. The same view applies even if the provider outsources its infrastructure to the cloud (indeed all publicly listed pod providers do so). Cybersecurity standards mandate a CISO should have access to data logs in order to respond to technical incidents.

Logging is a key mechanism according to ISO 27001 for obtaining snapshots of a system in order to detect possible sources of breaches and to improve security policies and solutions in place. In addition, in accordance with GDPR, logs may provide evidence in the event that concerns about access are raised (see also Req_09). Moreover, ISO 27701 prescribes the improvement of the security logging system to handle privacy. Logging systems need to be properly protected from fraudulent uses by external and internal actors. In addition, logging should be stored in a proper manner so as to be valid for use in a court of law. This means the logs should be tamper-proof, either by cryptographic means or by use of a trusted third party, so as to assert their trusted status if used as evidence in a court of law. Also, a proper authentication and authorisation solution should be put in place to discriminate who can have access to what kind of logs. Currently, the Solid specification does not define a logging system.

4.5. Identification of Purpose and Legal Basis (Req_05)

The Solid protocol suggests that users have control of their data by directly managing WAC and ACP. The intention is that data subjects, who store their data in their own pod, can determine for themselves whether an entity requesting to access the data has a *legal basis* and *purpose* to use the data in a particular way. The limitations of such arguments are that, firstly, manipulating WAC and ACP is too low-level for most users, and secondly, the relevant legal information is inferred from the external context, independently of Solid.

As mentioned in Section 3.3.6, controllers are obliged to record specific information related to processing activities. For large organisations and public entities, the controller assigns a DPO for whom recording such processing activities is one of their formal duties. For example, the DPO of a university records in an internal information system the purpose and legal basis for processing activities related to each ongoing research project involving

personal data. The question here is which agents in the Solid ecosystem are responsible for identifying the purpose and legal basis for processing personal data. The majority of data subjects will not have the legal expertise to make a judgement about what purpose and legal basis applies in a scenario. Ultimately, the responsibility for identifying the purpose and the legal basis is the controller responsible for the data user logging into an app, or the controller responsible for the app itself, depending on the use case. Thus, responsibility for a failure to record the correct purpose or legal basis for access by a data user at the time of each authorisation request lies with their controller, whether or not the data user followed correctly the guidelines recorded by their controller. Thus, by a pod requiring that the data user indicates their legal basis at the time of an authorisation request, the data subject has stronger grounds for holding the data user accountable, via their controller, in the case of a dispute.

Technical measures to address this requirement currently being explored include where ACP is extended to take into account for what purpose the personal data of a data subject is being processed [52–55]; that is, an access control policy in ACP should record explicitly the legal basis under which agents and apps are granted access to a resource. The purpose can also be recorded in specific accesses granted, and if an access control log is retained, recording the history of accesses granted, then we obtain a history of the purposes and legal basis associated with each grant. Such steps are "encouraged" by GDPR (cf. Recital 100), "allowing data subjects to quickly assess the level of data protection of relevant products and services".

In addition to recording the policy in the authorisation server of the pod, authorisation requests from Solid apps can also record the reason why the access is requested by making explicit the purpose in the scope of the ID token agreed upon with the agent logging in. This will explain how data users will process data upfront. The Data Protection Vocabulary (DPV) is a key step towards presenting such structured information so that the purpose may be taken into account by automated agents [56]. For example, personal data accessed may be used once in a computation to find the percentage of recovered COVID patients in a community, without revealing an individual's health status. For *data retention* purposes, which is another requirement of GDPR, it would be beneficial if the policy stipulates whether the data must be destroyed after use by the Solid app, or whether the data is permitted to be used only within a particular time window.

There are several possible mappings from controllers to concepts in Solid, as discussed in Section 3.1. For example, it is natural for controllers responsible for multiple data users (e.g., their employees) to act as an issuer in authentication protocols, as discussed in Section 4.1. This way, since the WebID of controllers would then become cryptographically tied to the ID token, then generic information that must be recorded, such as the contact details of the relevant DPO, can be recorded in the WebID. Similarly, if the processing activities of the app itself are of concern, the WebID of the app can record contact details of the relevant controller, and that WebID becomes cryptographically tied to a particular access. Since controllers can delegate some processing activities to processors that act on their behalf, but they remain responsible for them according to GDPR, authentication protocols could be adapted to provide the means for delegation of access to a processor (e.g., when a controller delegates part of its processing to a privacy-preserving service operated by another organisation). Authenticating chains of delegation is out of the scope of ACP and Solid OIDC currently, and would require more clarification than just making use of the `acl:delegates` term since such a pattern should be reflected in the authentication protocol.

4.6. Access Logs Recorded as Evidence of Policies and Accesses Granted (Req_06)

We now focus on access logs relevant to GDPR in the sense explained in Section 3.3.3. Access logs concern access to a pod of a data subject by data users, and their corresponding authorisation requests that established the policy under which a particular access for the access was granted. Records in logs of authorisation servers and Solid apps can build on ACP [21], which can describe access policies and grants recording, for example, the user,

app, and issuer used to log in; the resources accessed; operations granted; and the agreed terms of the policy granting access (see Req_04). Logs should also be retained by issuers to record the app, user, and scope of ID tokens approved.

To back up log entries with evidence, there are several aspects of each successful access that can also be logged by various actors in the Solid ecosystem. Assuming Solid OIDC is employed, the following cryptographic evidence is generated for each successful access request.

- The Solid app can log instances of issuance of an ID token as cryptographic proof of access being granted by the user via an issuer.
- The issuer can record evidence of their contribution to authentication, e.g., ID tokens they have issued in Solid OpenID Connect (or the verifiable credential they issue in VC protocols).
- The authorisation server can log cryptographic proof of access grants via the ID token and DPoP token used in the authorisation process, where the DPoP token contains context information about the scope of the access requested, notably the URI and method and a hash of a public session key, which cryptographically proves possession, by the app, of the ID token. The ID token cryptographically asserts the user, app, and issuer. The authorisation server may also log access tokens issued.
- The resource server can log access tokens used to access a resource.

The desired formal property we wish to rely on in order for the above evidence to serve as trustworthy evidence that can be used in logs to assert that the agents involved really approved is known as *non-repudiation* [57]; that is, by some agent presenting their logged evidence, it is impossible for the other agents involved in the protocol to deny that they were actually involved in their part of approving the access under the terms agreed.

What is notable about the above observations concerning the trustworthy evidence available to the various agents for logging purposes is that data users, subject to GDPR as controllers or processors, do not directly obtain cryptographic evidence of their accesses by virtue of participation in the Solid OIDC flow, according to the current protocol specifications. This suggests that data users are reliant on the logs retained by apps, pods and issuers that they use for cryptographic proof of access when the Solid OIDC flow is employed. The current flow proposed for Solid OIDC therefore requires the data user to trust one or more of these actors to properly handle accesses. Without relying on other actors, the data user may only record informally non-cryptographically asserted logging attempts to a particular app, issuer and scope, in the personal informal ledger of the data user, external to the Solid ecosystem. This is of course consistent with GDPR since the trust of logs may be inferred from the trustworthiness of the controller responsible for the data user; however, trust is enhanced in the case of legal disputes that bring into question the integrity of the data user if they can provide cryptographic non-repudiable proof to back up log entries.

In order to improve trust in the logs of data users, independently of an app they log into, a potential future solution is to support authentication flows, whereby the data user is involved directly in signing ID tokens (we cover further VC protocols with this property in Section 4.8). Going further, in order for a data user to observe accesses made by an app on their behalf, the data user may be involved in signing authorisation requests initiated by the app, where access is granted to a resource, as suggested in Figure 7. Thereby, each party receives cryptographic evidence of the agreement between the authorisation server, app, and data user, as part of the protocol, which they can independently log. If designed appropriately, such a flow can also provide an alternative resolution for some of the authentication concerns in Section 4.1 since the app is forced to check with the data user for each request to the authorisation server, permitting the data user to verify the access against their own policy for how the app may access data on behalf of the data user. This flow could be preferred by a data user if the data user does not fully trust the app to retain evidence and access logs (or indeed if the data user wishes to restrict the app). We are not aware of any such flow proposed for Solid.

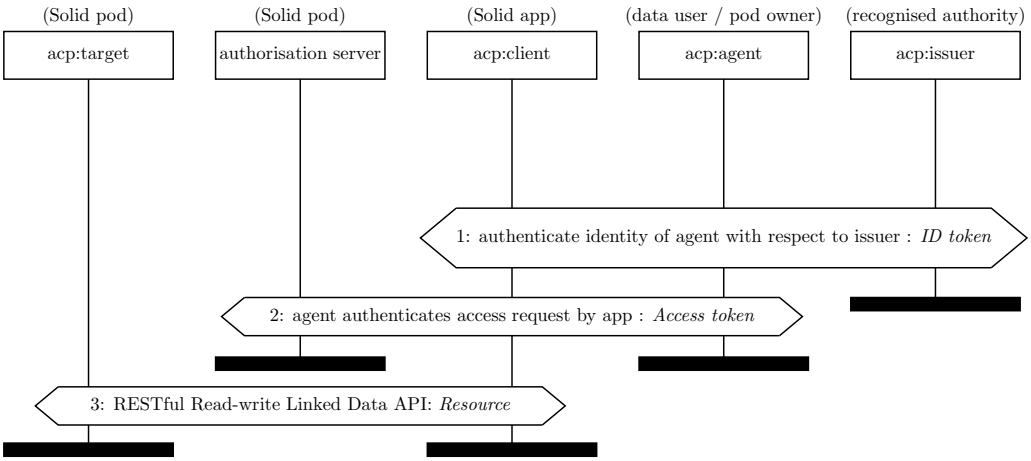

Figure 7. Alternative flow where the agent logged into an app is involved in the authentication of authorisation requests.

Another notable observation concerning the availability of authenticated cryptographic evidence is that there is nothing in the Solid specifications that cryptographically ties the access token used to access a resource via a resource server to a particular DPoP token where access to the resource was requested by the authorisation server. Hence, the authorisation server must be trusted to correctly assert the pairing of access token and resource access instance if it is required to provide cryptographic evidence of fine-grained access information. Specifically, the resource server may record that an access was made by an app to a particular resource with a given operation at a particular moment, and not just that the access was granted. Since access tokens are not specified, currently, the resource server must be trusted in order to believe that an access token was indeed used to access a particular resource, if the specific operations performed by an app are challenged. This means that access grants logged by the authorisation server currently contain the most pertinent trustworthy information about access grants, rather than the accesses to the resource server itself. Thus currently, having retained authorisation logs, one can prove that a user was allowed to access a resource but not that they did access a resource, which could be problematic legally regarding conflicts of interest for instance.

Recall that ACP separates specific instances of grant-access grant graphs from the policy—the authorisation graph. It is therefore appropriate to retain two separate logs for the authorisation server, where the log containing policy entries (authorisation graphs) can be used to provide explanations of each instance of an access grant under some policy. Having the policies logged separately, would permit replaying access grant requests in the past to explain, not only currently granted accesses, but also patterns of access in the past, which could be useful for explaining abuses of contractual agreements. An alternative to retaining the policy could be to rely on relevant policy information being recorded in each access grant graph, which has the advantage that less information about a policy need be leaked if the policy employed is of a general nature that pertains to a broader context than requested, e.g., if the policy encompasses conditions under which other data users may access the data.

ACP can be extended with features that can help usage control. At a basic level, ACP does not currently specify how timing information is recorded, which would be included in access logs, but would be better still built into policies, as we explain. For policies, the invocation timestamp and expiry of a general policy can be useful to limit processing activities and for logging purposes. Also, recording the duration of instances permitted by a policy allows the expiry of each access grant instance to be set automatically in the resulting access grant graph. Thus access instances, recorded by access grant graphs, should record their time validity and revocation conditions. As with other policy information, this should also be authenticated via the authentication protocol employed, e.g., backed up cryptographically via appropriate timestamps in ID tokens and DPoP tokens.

Going further, the access token issued by an authorisation server may be stateful and build in a particular control of usage, prescribing a process with specific URIs that may be accessed and operations that may be performed on them in a particular order.

By logging policies, non-compliance with obligations in such policies can be detected [58,59]; however, proposals related to usage control are not yet incorporated into Solid specifications. Ensuring policies are logged caters for such future extensions to Solid, in addition to improving accountability in the short term.

As mentioned at the beginning of this section, ideally, logs should be supported cryptographically, and some information, but not all such information, is provided by Solid OIDC. Solid OIDC was never designed with producing cryptographic non-repudiable evidence with logs in mind. Thus, the observations we make are a creative step to leverage the existing technology to support logging. A better solution would be to design the authentication protocols and the certificates they produce with the logging consideration we have mentioned in mind.

An alternative to producing trusted cryptographic evidence of logging is to use access logs simply asserting the relevant context graph, and for the access logs to be governed by a trusted third party. The obvious incarnation of this is that the authorisation server may be governed by a pod provider independently of the pod owner, meaning that logs presented as evidence of an access by the authorisation server can be trusted independently of claims made by the pod owner or data user, assuming that the pod provider is legally external to the dispute and makes use of a trusted solution such as an encrypted database that they do not access themselves (in particular the pod cannot be self-hosted by the pod owner). Whether using cryptography or a trusted 3rd party, either approach to tamper-proofing access logs should guard against scenarios where a malicious data subject aims to incriminate a data user by claiming falsely they accessed the system in violation of an agreed purpose, and hence the data subject owning the pod should not be able to forge a proof of an access that did not occur.

Related work by Pandit [60] on GDPR and Solid also emphasises the need for different types of logs to be retained by different actors in the Solid ecosystem. Pandit points out correctly that the Solid interoperability draft [50] does introduce the notion of an access receipt that an agent can provide to another agent. This may be useful for presenting evidence of access to users for example. However, as observed by Pandit, the form of access receipts is not specified. Going further, we add that just providing a receipt as a message does not mean the receipt can be trusted and thus such a receipt should be part of the authentication protocol, and properties such as non-repudiation of the receipt should be verified if the receipt is to be trusted as proof in the relevant contexts.

4.7. Multi-Factor Authentication (Req_07)

User account management is well known to be one of the primary sources of data breaches. At the time of writing, Solid only requires the WebIDs of actors involved in the ecosystem and a password from the agent logging in to identify themselves, regardless of whether they are data users, pod owners, or DPOs, and regardless of what operations they perform. WebIDs are currently public and passwords are vulnerable to brute-force attacks and can potentially be stolen or leaked to other parties. To decrease the risk of successful cyberattacks or data breaches, multi-factor authentication, which requires two or more verification factors, can be applied. For example, two-factor authentication, where we use a password and some other source of authentication, such as a text message, can be built into an issuer used to authenticate access to a pod. Multi-factor authentication is particularly important for account initialisation, account recovery, and critical actions such as accessing a view of access control logs by a CISO. Multi-factor authentication is not recommended in Solid OIDC yet; however, including it explicitly could increase confidence in Solid. If a pod owner grants access to a data user with whom the pod owner has never interacted, then multiple factors may be authenticated, such as the proximity of a data user if they are physically near to the pod owner, e.g., a doctor in a clinic. Some new authentication

methods that are integrated with machine learning models can authenticate users based on their locations, such as geo-locations or IP addresses or based on the context or behaviour according to users' regular behaviour. A range of new authentication methods would strengthen the security of Solid account management.

4.8. Unlinkability to Protect against External Profiling (Req_08)

In the scope of Solid is the privacy property *unobservability* [39], which ensures that certain operations cannot be observed by a man in the middle. If some functionalities are outside the pod, such as a wallet containing logs proposed in related work [55], there is the unnecessary risk of exposing security-critical data over the internet. Thus, critical operations are better made internal to the pod, hence unobservable with respect to man-in-the-middle attacks on the network. Of course, some observability must be tolerated in order for Solid to fulfil its philosophical role. One could argue that in the traditional data processing model, in Figure 4, where the data user stores the data of the data subject, there is better privacy since unobservability is controlled by the internal organisational security measures of the data user. In a data-sovereignty scenario, in Figure 3, involving a pod controlled by the data subject, rather than the data user, there is no possibility of a hardware solution that makes all operations unobservable.

When we do not have unobservability, the strongest privacy property we can target is *unlinkability* [39], discussed already in terms of data minimisation requirements in Section 3.3.7. This ensures that two uses of the same service cannot be linked by actors external to the given sessions of the protocol; thus, it is dependent on the perspective of who is performing the linking. Unlinkability is violated for Solid OIDC from the perspective of the issuer since the issuer may trivially connect two authentication attempts by the same user. Since an issuer can learn over the time that users access pods, this may expose sensitive information that may be used to profile the data subject, such as the processing activities a data subject is subject to, thereby impacting the privacy of the data subject. We explain next how emerging technology, under consideration in the Solid ecosystem, may address this.

Unlinkable Alternatives to Solid OIDC for Authentication via Verifiable Credentials

The Solid specification is expected to evolve to permit alternatives to Solid OIDC for authentication. Solid OIDC, or indeed OpenID Connect in general, may also raise legal obligations in terms of data protection since the issuer can trace which relying parties the user logs in to, which is a power given to the issuer that is routinely monetised and quite opposed to the principle of data sovereignty [61]. While it is acceptable for the credentials of the data subject to be revealed to their own pod, a data subject may not wish to reveal their behaviours to an issuer. Indeed, even exposing information flows concerning the CISO to an issuer may reveal unwanted information about the business model of the pod provider, for example.

The above may be avoided by using a verifiable credential-based authentication protocol, where the issuer signs attributes for authentication that prove that the data subject owns the pod. The W3C is taking steps in this direction via the DID and verifiable credential recommendations [17,18]. However, these official recommendations do not specify protocols, and such a protocol should be verified since man-in-the-middle attacks and privacy attacks may be introduced without careful design of security controls in the protocol. A verifiable credential-based authentication mechanism is an ongoing subject of discussion with respect to Solid. A suitably designed [62,63] VC protocol should satisfy unlinkability from the perspective of the issuer, in the sense that having issued the credential, it is impossible for the issuer to link two sessions where credentials are used, assuming the data user and app are honest, and even when the session involves a credential that it issued. A typical flow for an unlinkable VC protocol is presented in Figure 8, where the information typically provided in an ID token is authenticated in two stages, as suggested by the attributes struck out. An experimental VC-based protocol is being aired in Inrupt's Enterprise Solid Server,

which is, however, designed such that the holder of credentials is not protected from the issuer, and thus unlinkability from the perspective of the issuer does not hold. Thus, that protocol appears to use VCs to enhance data interoperability and accountability rather than unlinkability.

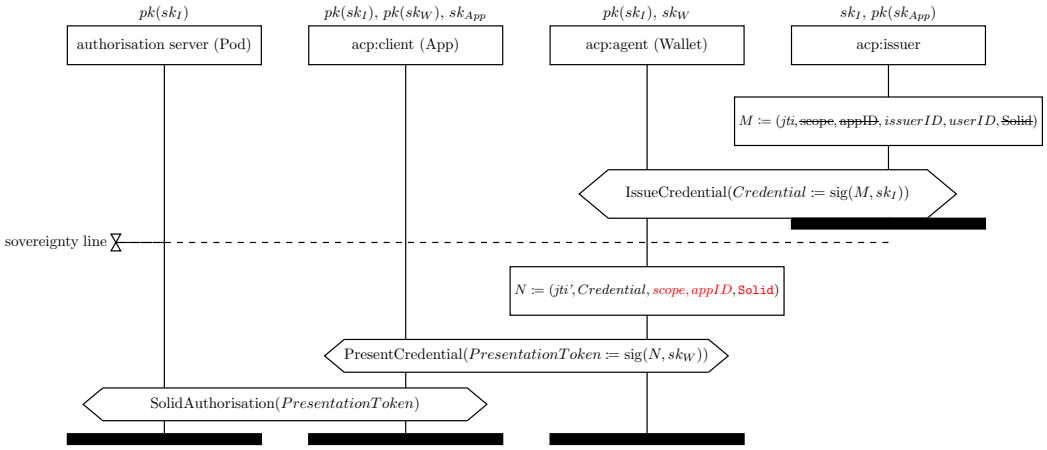

Figure 8. Illustration of the flow of a VC-based protocol, where the issuer is not involved in either authenticating the user with an app of authorisation, yet much of the functionality of Solid OIDC is preserved. The sovereignty line emphasises that information below the line does not flow to the issuer.

4.9. Protocols Facilitating the Rights of Data Subjects (Req_09)

Article 20 enshrines the right to data portability, allowing the data subject to obtain their personal data in a structured format. Thus, Article 20 is catered for by using Solid to store personal data, as in the use cases we focus on throughout this paper.

Articles 16–18 of GDPR enshrine the right of data subjects to rectify their personal data, erase their data entirely, or restrict their usage for data processing. By Article 19, the controller has the responsibility to notify the data subject of compliance with such requests. We discuss here how such rights may be supported by an additional protocol layer involving the controller and the pod.

In Solid, one may argue that since data subjects store their personal data in their own pod and control access, they have direct control of these processes. In the current version of Solid, the data subject always has full control of the data in their own pod, which means they are always able to write, modify, and remove data elements and files in their pods. A limitation with such an argument is that there are scenarios where erasing the data stored in the pod does not mean that the data is erased by data users connected to the pod since the data user also retains a copy, and the data in the pod is a courtesy to the pod owner for transparency purposes and to leverage Solid to facilitate compliance. Going further, there are scenarios where a contractual agreement does not allow the data subject to modify, remove, or restrict data stored in their own pod freely, even if they have full power to inspect the data, access control policies, and logs. One such scenario may be if data is required for billing purposes or medical records. In other scenarios, data subjects may prefer to permit management of part of their personal data to be handled most of the time by the relevant data user since they may lack the expertise or time to understand their data and modify them correctly.

In any of the above-mentioned situations, we propose that a protocol involving the data subject owning the pod and the controller responsible for data users would facilitate compliance with Article 19. Such a protocol would notify the controller about requests to erase (via an app with suitable access to the data concerned, for example), in order to facilitate the erasure of personal data across multiple locations, not only the pod. Such a protocol would also enable the controller to explain why certain requests for erasure may

not be possible. This is likely an additional protocol layer built on top of the existing Solid specifications, rather than an enhancement of the existing REST operations.

Similarly to erasure, the right to restrict processing may only be triggered via Article 19 and not unilaterally by the data subject without notification of the request to change the purpose of processing. It would be incorrect just to change the purpose of processing in the access control policies of the pod unilaterally.

Article 21 ensures a data subject may always challenge processing activities. This is partly catered for by Solid, in that there is transparency about the data stored and the accesses, which can be used by the data subject to substantiate an objection. For example, it was reported (https://www.dublineconomy.ie/insights/american-football-game-linked-to-us-tourist-spending-surge-in-dublin-17703/ accessed 8 December 2022) that Mastercard partnered with Dublin Council to process personal data on payments during sporting events hosted by the city. If someone was in Dublin during events, they may be reassured by the fact that anonomysation was applied, but alternatively, they may wish to exercise their right not to be subject to such processing, or even question whether there was sufficient legal basis. In such scenarios, if the personal data held and the accesses to that data made by the organisation involved were made transparent by logging them through Solid, then resolutions according to the preferences of data subjects can be facilitated. Therefore, this right could be facilitated by agreeing on protocols for notifying a relevant controller about specific objections, and by the use of cryptography to ensure tamper-proof logs. Such a protocol should be able to produce evidence to back up a request from the data subject to the controller. This includes evidence recorded concerning accesses and the purpose of accesses extracted from access logs of entities wishing to comply, such an authorisation server retaining logs on behalf of the data subject, as discussed in Section 4.6.

Similar provisions may be built into Solid to facilitate the right enshrined in Article 22, concerning the right not to be subject to decisions based solely on automated processing. Article 22 is, of course, related to emerging regulation concerning the ethical use of AI.

4.10. Notification of Data Breaches (Req_10)

Both data access logs, recorded for GDPR purposes, and underlying network and system logs can be used by the pod provider to detect and assess potential technical incidents. There are currently no provisions for notification of data breaches in the Solid specification.

Personal Data Breaches. Article 34 of GDPR requires that personal data breaches must be reported by controllers whenever they occur—for instance, if data intended for processing is transmitted accidentally to the wrong people (in a misfired email for example) or a machine processing data is compromised. Such data breaches are reported to or by the controller responsible for the data user, who must communicate the data breach to the data subjects involved. Reporting of data breaches involving data stored in Solid pods can be facilitated via a suitable protocol, where the controller responsible for the data user makes use of their view of the access logs of a pod, discussed in Section 4.6. The data subject can then cross reference the relevant logs against their own logs in order to assess how they are personally impacted by the data breach.

Violations of Purpose. When a data subject has data available in their own pod, the data subject should be able to inspect manually the access logs to determine whether there has been an access for an illegitimate purpose. For instance, suspicions may be raised if credentials related to emergency services have been employed to access health data when the data subject was in fact healthy and not in need of care. This could occur due to the account of a legitimate data user becoming compromised, for instance. In practice, such manual inspection would be too low level for most users. However, a pod provider may provide, as a service, tools that analyse the access logs of a pod and summarise and explain at a high level potential irregular access patterns. Such a tool would require access to logs in order to automate this analysis, and the tool would need to be GDPR-compliant itself, ensuring that the access logs are used for the given purpose only.

Technical Incidents. In accordance with ISO 27001 with respect to cyberattacks and ISO 27701 for privacy-related incidents, reviewing the collected security-related data can be useful for the detection of possible violations, and organisations should establish responsibilities and procedures related to collecting information related to occurred issues and detected violations and their notification. It is reasonable that these can be automated by the pod or its provider. Such cyber-incident reporting could be provided as a service by the pod provider to the user, and may be based solely on logs external to the pod described in Section 4.4, for instance, based on internal datacentre traffic where the pod is hosted. This solution can be augmented according to ISO 27701 in order to realise a Privacy Information Management System for a more effective and efficient management of privacy concerns. These aspects are more related to pod providers and are not covered by the Solid specification.

5. Open Issues and Challenges

Besides the open technical challenges highlighted in the previous section concerning the evolution of the Solid specifications in order to improve GDPR compliance and tool support, there is also the issue of keeping up with emerging legislation initiatives. We present here some emerging legislation that we expect to impact Solid. We leave as a future work a more detailed analysis when the legislation is approved and their content is precisely known since for some of them only drafts are circulating.

First of all, the topic of data protection in the EU has been approached using a couple of legal tools: a so-called lex generalis, issuing a set of generic and universal rights and obligations, and a lex specialis, overriding the general provisions and issuing specific norms with respect to a particular context. The relevant lex generalis is of course GDPR 2016/679, which superseded the Data Protection Directive 95/46/EC (DPD). Connected to them, we have the Privacy and Electronic Communications Directive 2002/58/EC on Privacy and Electronic Communications or ePrivacy Directive (ePD), which has been amended by Directive 2009/136. The ePD directive is narrow in scope, targeting only the confidentiality of network-enabled communications, and the treatment of traffic data, spam, and cookies, and specialising in DPD. Therefore, the EU legislation is incomplete in the sense that a counterpart of GDPR has not been approved yet. In the near future, we expect to witness new regulations superseding the ePD, as GDPR has superseded DPD. This regulation, named the ePrivacy Regulation (ePR), will introduce important novelties that we will explain next.

A limitation of ePD is that it addresses traditional telecom operators, and not necessarily other players in the digital society and ICT market, which are subject to GDPR. Furthermore, GDPR obligations are ambiguous regarding how they are applied by specific actors. As evidence, there are ongoing debates on how blockchain, cloud, or IoT can be compliant with GDPR. This issue we expect to be addressed by ePR by prescribing that such technologies meet security and privacy requirements comparable to telecom operators who are already subject to ePD. This will be made possible by introducing the concept of *data intermediaries* that are involved in data processing on behalf of data users, and by specifying the legal obligations of such data intermediaries. We expect Solid pod providers and Solid app providers, for example, to be seen as intermediaries, and hence ePR will likely augment the requirements in this paper. For example, in ePR, consent gathering and handling is expected to be more user-friendly, and hence actors in the Solid ecosystem should gather, log, and manage consent according to this new model.

In addition to these legislative efforts to complete the EU data protection framework, novel legal tools related to cybersecurity and resiliency are emerging. The Cyber Resilience Act (CRA) aims to define common IT security standards for digital products connected to the network (so-called "IoT") and related services. This will add to the recent NIS 2 Directive 2022/2555 that has been approved to respond to the growing threats posed by digitisation and the wave of cyberattacks. These novel EU legislative initiatives and tools will strengthen and enlarge the set of requirements we have listed in Table 2, for which

actors in the Solid ecosystem will need to find solutions. Last but not least, the EU Council has also approved the Data Governance Act (DGA), applicable from 24 September 2023, which has been anticipated by elements of the Open Data Directive (2019/1024, ODD) concerning *data altruism* aimed at liberalising the data market. DGA aims to support a flourishing European data economy, but does not replace the rights and protections set in place by GDPR. New roles in DGA should be mapped to those in GDPR. Specifically, DGA defines novel types of data intermediaries we expect to be relevant to Solid. The role of data intermediaries as defined under DGA, together with an expansion of data subject's rights (more access control, transparency, and data portability) in the complementary Data Act (currently under review) seem to fit with principles underlying Solid. All of these legal aspects will surely give more business opportunities and growth to Solid and, at the same time, will impose novel challenges to be properly addressed, leading to an evolution of the standards and technologies governing the Solid project.

In light of these novelties, we expect Table 1 and Table 2 to evolve over time, along with technological and societal advancements and related forthcoming legislation to regulate them. In order to make this evolution feasible and effective from a practical point of view, we propose that such tables are published and maintained online thus allowing external contributors to also join the discussion and point out relevant literature and supporting documentation. In addition, they might be stored and visualised as semantically enriched hypertext that also employs AI to check consistency/completeness of the coverage of legal requirements (see [7,64] for the use of AI on similar use cases). The present paper can be a starting point for the creation of such an online repository and community.

6. Conclusions with Recommendation

In this paper, we elicited requirements from GDPR and officially approved documents, as summarised in Table 1 and how they relate to measures in the Solid protocol and its ongoing evolution, as summarised in Table 2. We also present, in Section 3.2, an overview of a Solid-based healthcare system developed by VITO, which concretely illustrates a typical use case where our legal analysis applies. Indeed, some of our analysis is necessary in order for the system to be deployed outside a sandbox where the healthcare data of real data subjects is processed, notably the controllers in the system and their responsibilities must be identified and catered for appropriately.

We reflect on challenges uncovered for strengthening privacy in Solid. Foremost, the access logs should be an integral part of Solid pods and Solid apps, as a record of accesses granted required for security and privacy audits and to support various rights of the data subject (c.f. Req_04, Req_06, Req_08, Req_10). Evidence can be used by data subjects who own a pod to challenge the behaviour of data users and apps using data in the pod, by notifying the relevant controller while providing a relevant view of the logs that they can back up with cryptographic evidence that the accesses concerned were indeed granted in a given context (Req_08, Req_10). Providing logs externally to a pod, in a separate wallet, creates unnecessary risks as more privacy-critical interactions are exposed over the internet than required; instead, each actor in the Solid ecosystem (authorisation servers, resource servers, Solid apps, data users) should log locally their view of interactions during authentication, along with cryptographic evidence derived from messages they send and receive during the authentication protocol itself. Such cryptographic evidence generated during the running of the authentication protocols used to log in and grant access to resources, elevates trust in logs. This way, controllers responsible for accesses violating their stated purpose may be held accountable, without the ability of the controller to cast doubt on the records of the data subject (Req_10). To support this, both access control policies and authentication protocols should be aligned on context information, including the purpose of an access (Req_01, Req_05). Indeed, failing to align context information in the authentication and access control mechanisms appropriately can leave Solid open to data breaches as explained in Section 4.1. Current low-level APIs for granting access leave room for developers, who need not be privacy experts, to make mistakes.

This work also makes a case for Solid to support more normative authentication protocols beyond Solid OIDC (c.f., Req_01, Req_04, Req_05, Req_06, Req_07, Req_08). The current normative proposal, Solid OIDC, is vulnerable to attacks violating authentication, although readily implementable measures are proposed to address those vulnerabilities, as explained when discussing Req_01. The greater challenge to address is that Solid OIDC cannot be upgraded directly to support data minimisation via unlinkability from the perspective of an issuer, as explained in the discussion surrounding Req_09. This leads us to suggest that Solid should support verifiable-credential-based protocols [17], but warn that such protocols should be carefully designed so that they indeed support unlinkability towards issuers, generate non-repudiable cryptographic evidence for parties involved in the protocol (see Section 4.6), align context information with ACP to avoid attacks (c.f. Req_01), support alternative flows (c.f. Figure 7), support delegation patterns for processors acting on behalf of controllers, etc. In addition to introducing new normative protocols, stronger multi-factor authentication methods should also be supported, e.g., for administrative functions (c.f. Req_07).

Additional protocols between data subjects and controllers, external to those already defined in the Solid specifications, can be introduced to explicitly facilitate the rights of data subjects and data breach notifications (c.f. Req_06, Req_09), and to streamline the compliance of controllers themselves. Such a layer would enhance Solid as a tool for facilitating compliance with GDPR, but is, thus far, not part of the considerations of the Solid ecosystem.

There are of course many privacy obligations that perhaps should not be catered for by the Solid protocol, but instead lie with pod providers. For example, additional physical and organisation measures typical of ISO 270001 should be enforced by pod providers (c.f. Req_03). We quote GDPR Recital 100, which is pertinent to Solid:

> "In order to enhance transparency and compliance with this Regulation, the establishment of certification mechanisms and data protection seals and marks should be encouraged."

This suggests that the degree of GDPR compliance should be transparently asserted in relation to the technological solution. By tightening specifications we tighten all implementations, and can perhaps facilitate steps towards identifying what certification and seals and marks are appropriate for technologies and actors diligently adhering to standards. One may also leverage the analysis in this paper to prioritise considerations towards certification of such actors in the Solid ecosystem.

Finally, we emphasise that the security and privacy of any system is a moving target since new vulnerabilities are discovered in key standards and libraries on a regular basis. Some of the suggestions in this work (e.g., tightening Solid OIDC using RFC 9207 in Req_01) illustrate this kind of evolution. A possible path for addressing this is to separate the core Solid protocol from an evolving security and privacy review that is updated as vulnerabilities are disclosed. This suggestion can be seen as an evolution of the Security and Privacy Review section in the draft Solid protocol at the time of writing [2], which is a list covering generic self-review questions for Web platforms [65]. Such a review can serve as a policy benchmark for pod providers, app providers, and other developers to adhere to, thereby improving trust in the Solid ecosystem.

Author Contributions: Writing—original draft preparation, C.E., R.H., L.R., B.B. and E.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the COST Action on Distributed Knowledge Graphs (CA19134), supported by COST (European Cooperation in Science and Technology).

Acknowledgments: We thank Olaf Hartig, Chang Sun, and Efstratios Koulierakis for their discussions on the topic. This work was stimulated by two workshops on Privacy Issues in Distributed Social Knowledge Graphs held, respectively, at the University of Luxembourg, 13–15 June 202, reported on in an earlier draft of this paper [66], and the University of Salerno, 13–15 February 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sambra, A.V.; Mansour, E.; Hawke, S.; Zereba, M.; Greco, N.; Ghanem, A.; Zagidulin, D.; Aboulnaga, A.; Berners-Lee, T. *Solid: A Platform for Decentralized Social Applications Based on Linked Data*; Technical Report; MIT CSAIL & Qatar Computing Research Institute: Cambridge, MA, USA, 2016.
2. Capadisli, S.; Berners-Lee, T.; Verborgh, R.; Kjernsmo, K. Solid Protocol, 2023. Version 0.11.0, Editor's Draft. Available online: https://solidproject.org/ED/protocol (accessed on 12 July 2023).
3. Ajani, G.; Boella, G.; Di Caro, L.; Robaldo, L.; Humphreys, L.; Praduroux, S.; Rossi, P.; Violato, A. The European legal taxonomy syllabus: A multi-lingual, multi-level ontology framework to untangle the web of European legal terminology. *Appl. Ontol.* **2016**, *11*, 325–375. [CrossRef]
4. Robaldo, L.; Villata, S.; Wyner, A.; Grabmair, M. Introduction for artificial intelligence and law: Special issue "natural language processing for legal texts". *Artif. Intell. Law* **2019**, *27*, 113–115. [CrossRef]
5. Robaldo, L.; Antoniou, G.; Baryannis, G.; Batsakis, S.; Governatori, G.; Islam, M.B.; Liu, Q.; Siragusa, G.; Tachmazidis, I. Large-scale Legal Reasoning with Rules and Databases. *J. Appl. Log.* **2021**, *8*, 911.
6. Bartolini, C.; Giurgiu, A.; Lenzini, G.; Robaldo, L. Towards Legal Compliance by Correlating Standards and Laws with a Semi-automated Methodology. In Proceedings of the BNAIC 2016: Artificial Intelligence, Amsterdam, The Netherlands, 10–11 November 2017; Volume 765, pp. 47–62. [CrossRef]
7. Robaldo, L.; Bartolini, C.; Palmirani, M.; Rossi, A.; Martoni, M.; Lenzini, G. Formalizing GDPR provisions in reified I/O logic: The DAPRECO knowledge base. *J. Log. Lang. Inf.* **2020**, *29*, 401–449. [CrossRef]
8. EU Cloud. EU Cloud Code of Conduct. 2020. Available online: https://eucoc.cloud/en/home (accessed on 12 July 2023).
9. CISPE. Data Protection Code of Conduct for Cloud Infrastructure Service Providers. 2021. Available online: https://cispe.cloud/code-of-conduct/ (accessed on 12 July 2023).
10. CNPD. GDPR-CARPA (Version 1/2022), 2022. Decision 15/2022. Available online: https://cnpd.public.lu/dam-assets/fr/professionnels/certification/decision-n-15-2022-du-13-mai-2022-criteres-de-certification.pdf (accessed on 13 May 2022).
11. Article 29 Data Protection Working Party. Opinion 5/2009 on Online Social Networking. 01189/09/EN WP 163. 2009. Available online: https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2009/wp163_en.pdf (accessed on 12 June 2009).
12. Coburn, A.; Pavlik, E.; Zagidulin, D. SOLID-OIDC, 2023. Editor's Draft. Available online: https://github.com/solid/solid-oidc (accessed on 7 March 2023).
13. Sakimura, N.; Bradley, J.; Jones, M.; De Medeiros, B.; Mortimore, C. *OpenID Connect Core 1.0.*; Technical Report; The OpenID Foundation: San Ramon, CA, USA, 2014.
14. Faísca, J.G.; Rogado, J.Q. Decentralized semantic identity. In Proceedings of the SEMANTiCS 2016: 12th International Conference on Semantic Systems, Leipzig, Germany, 12–15 September 2016; pp. 177–180. [CrossRef]
15. Oraskari, J.; Törmä, S. Access control for Web of building data: Challenges and directions. In *eWork and eBusiness in Architecture, Engineering and Construction*; CRC Press: Boca Raton, FL, USA, 2017; pp. 45–53.
16. Story, H.; Corlosquet, S.; Sambra, A. WebID-TLS: WebID Authentication over TLS. Technical Report, W3C, 2014. Editor's Draft. Available online http://www.w3.org/TR/auth-webid/ (accessed on 5 March 2014).
17. Sporny, M.; Noble, G.; Longley, D.; Burnett, D.C.; Zundel, B.; Hartog, K.D. (Eds.) Verifiable Credentials Data Model v1.1. 2022. W3C Recommendation. Available online: https://www.w3.org/TR/vc-data-model/ (accessed on 3 March 2022).
18. Sporny, M.; Guy, A.; Sabadello, M.; Reed, D. Decentralized Identifiers (DIDs) v1.0, 2021. W3C Recommendation. Available online: https://www.w3.org/TR/did-core/ (accessed on 19 July 2022).
19. Sacco, O.; Passant, A.; Decker, S. An access control framework for the Web of Data. In Proceedings of the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Changsha, China, 16–18 November 2011.
20. Capadisli, S.; Berners-Lee, T. Web Access Control, 2022. W3C Candidate Recommendation, Version 1.0.0, Editor's Draft. Available online: https://solidproject.org/TR/wac (accessed on 5 July 2022)
21. Bosquet, M. Access Control Policy (ACP), 2022. Solid Editor's Draft. Available online: https://solid.github.io/authorization-panel/acp-specification/ (accessed on 20 September 2022).
22. Sandhu, R.S.; Samarati, P. Access control: Principle and practice. *IEEE Commun. Mag.* **1994**, *32*, 40–48. [CrossRef]
23. Speicher, S.; Arwe, J.; Malhotra, A. Linked Data Platform 1.0. Technical Report, 2015. W3C Recommendation. Available online: https://www.w3.org/TR/ldp/ (accessed on 26 February 2015).
24. Horne, R.J. Programming Languages and Principles for Read–Write Linked Data. Ph.D. Thesis, University of Southampton, Southampton, UK, 2011.
25. Berners-Lee, T.; O'Hara, K. The read–write linked data web. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2013**, *371*, 20120513. [CrossRef] [PubMed]
26. Stadtmüller, S.; Speiser, S.; Harth, A.; Studer, R. Data-fu: A language and an interpreter for interaction with read/write linked data. In Proceedings of the 22nd international conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 1225–1236.

27. Capadisli, S. Solid Notifications Protocol. Technical Report, 2023. Editor's Draft. Available online: https://solid.github.io/notifications/protocol (accessed on 21 June 2023).
28. Capadisli, S.; Guy, A.; Lange, C.; Auer, S.; Sambra, A.; Berners-Lee, T. Linked data notifications: A resource-centric communication protocol. In Proceedings of the The Semantic Web: 14th International Conference, ESWC 2017, Portorož, Slovenia, 28 May–1 June 2017; pp. 537–553.
29. Fox, A.; Griffith, R.; Joseph, A.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; Rabkin, A.; Stoica, I. *Above the Clouds: A Berkeley View of Cloud Computing*; Electrical Engineering and Computer Sciences University of California at Berkeley: Berkeley, CA, USA, 2009.
30. Mansour, E.; Sambra, A.V.; Hawke, S.; Zereba, M.; Capadisli, S.; Ghanem, A.; Aboulnaga, A.; Berners-Lee, T. A demonstration of the Solid platform for social Web applications. In Proceedings of the Companion Proceedings of the 25th International Conference on World Wide Web, Montreal, QC, Canada, 11–15 April 2016.
31. Verbrugge, S.; Vannieuwenborg, F.; Van der Wee, M.; Colle, D.; Taelman, R.; Verborgh, R. Towards a personal data vault society: An interplay between technological and business perspectives. In Proceedings of the 60th FITCE Communication Days Congress for ICT Professionals, Vienna, Austria, 29–30 September 2021. [CrossRef]
32. McKean, R.; Kurowska-Tober, E.; Waem, H.; de Souza, R. GDPR Fines and Data Breach Survey: January 2023. Technical Report, DLA Piper, 2023. Available online: https://www.dlapiper.com/en-ae/insights/publications/2023/01/dla-piper-gdpr-fines-and-data-breach-survey-january-2023 (accessed on 12 July 2023).
33. EDPB. Guidelines 07/2020 on the Concepts of Controller and Processor in the GDPR (Version 2.1), 2020. Available online: https://edpb.europa.eu/our-work-tools/documents/public-consultations/2020/guidelines-072020-concepts-controller-and_en (accessed on 2 September 2020).
34. Janssen, H.; Cobbe, J.; Norval, C.; Singh, J. Decentralized data processing: Personal data stores and the GDPR. *Int. Data Priv. Law* **2021**, *10*, 356–384. [CrossRef]
35. Fabbrini, F.; Celeste, E. The right to be forgotten in the digital age: The challenges of data protection beyond borders. *Ger. Law J.* **2020**, *21*. [CrossRef]
36. ECJ. Case of Wirtschaftsakademie Schleswig-Holstein GmbH (C-210/16), 2018. Available online: https://curia.europa.eu/juris/liste.jsf?num=C-210/16 (accessed on 5 June 2018).
37. EDPB. Guidelines 9/2022 on Personal Data Breach Notificiation under GDPR, 2022. Available online: https://edpb.europa.eu/our-work-tools/our-documents/guidelines/guidelines-92022-personal-data-breach-notification-under_en (accessed on 28 March 2023).
38. EDPB. Guidelines 09/2020 on Relevant and Reasoned Objection under Regulation 2016/679 (Version 2.0), 2021. Available online: https://edpb.europa.eu/our-work-tools/documents/public-consultations/2020/guidelines-092020-relevant-and-reasoned_en (accessed on 8 October 2020).
39. National Security Agency. *Common Criteria for Information Technology Security Evaluation*; National Security Agency: Fort Meade, MD, USA, 2017.
40. Bao, D.; Miura, J.; Zhang, N.; Goto, Y.; Cheng, J. Supporting verification and validation of security targets with ISO/IEC 15408. In Proceedings of the 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), Shengyang, China, 20–22 December 2013; pp. 2621–2628.
41. Nissenbaum, H. Privacy as contextual integrity. *Wash. L. Rev.* **2004**, *79*, 119.
42. Anderson, J.M. Why we need a new definition of information security. *Comput. Secur.* **2003**, *22*, 308–313. [CrossRef]
43. *ISO/IEC 27001: 2017-06*; Information Security Management Systems. International Organization for Standardization & International Electrotechnical Commission Std.: Geneva, Switzerland, 2017.
44. Lachaud, E. ISO/IEC 27701 Standard: Threats and Opportunities for GDPR Certification. *Eur. Data Prot. L. Rev.* **2020**, *6*, 194. [CrossRef]
45. Das, A.; Bonneau, J.; Caesar, M.; Borisov, N.; Wang, X. The tangled web of password reuse. In Proceedings of the NDSS, San Diego, CA, USA, 23–26 February 2014; Volume 14, pp. 23–26.
46. Morgan, J.; Coburn, A.; Bosquet, M. Solid-OIDC Primer, 2023. Editor's Draft. Available online: https://solid.github.io/solid-oidc/primer/ (accessed on 7 March 2023).
47. Li, W.; Mitchell, C.J. Analysing the Security of Google's Implementation of OpenID Connect. In Proceedings of the Detection of Intrusions and Malware, and Vulnerability Assessment: 13th International Conference, DIMVA 2016, San Sebastián, Spain, 7–8 July 2016; pp. 357–376.
48. Fett, D.; Küsters, R.; Schmitz, G. A Comprehensive Formal Security Analysis of OAuth 2.0. In Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security. Association for Computing Machinery, Vienna, Austria, 24–28 October 2016; pp. 1204–1215. [CrossRef]
49. OAuth 2.0 Authorization Server Issuer Identification. Standards Track 9207, 2022. Available online: https://datatracker.ietf.org/doc/rfc9207/ (accessed on 18 March 2022).
50. Solid Application Interoperability. Technical Report, W3C Solid Community Group, 2023. Editor's Draft. Available online: https://solid.github.io/data-interoperability-panel/specification/ (accessed on 4 April 2023).
51. Chen, S.; Wang, R.; Wang, X.; Zhang, K. Side-Channel Leaks in Web Applications: A Reality Today, a Challenge Tomorrow. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 16–19 May 2010. [CrossRef]

52. Havur, G.; Sande, M.V.; Kirrane, S. Greater Control and Transparency in Personal Data Processing. In Proceedings of the 6th International Conference on Information Systems Security and Privacy (ICISSP), Valletta, Malta, 25–27 February 2020. [CrossRef]
53. Esteves, B.; Pandit, H.J.; Rodriguez Doncel, V. ODRL Profile for Expressing Consent through Granular Access Control Policies in Solid. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Vienna, Austria, 6–10 September 2021; pp. 298–306. [CrossRef]
54. Esteves, B.; Rodríguez-Doncel, V.; Pandit, H.J.; Mondada, N.; McBennett, P. Using the ODRL Profile for Access Control for Solid Pod Resource Governance. In Proceedings of the The Semantic Web: ESWC 2022 Satellite Events, Hersonissos, Greece, 29 May–2 June 2022; Volume 13384, pp. 16–20. [CrossRef]
55. Debackere, L.; Colpaert, P.; Taelman, R.; Verborgh, R. A Policy-Oriented Architecture for Enforcing Consent in Solid. In Proceedings of the 2nd International Workshop on Consent Management in Online Services, Networks and Things, Lyon, France, 25–29 April 2022; pp. 516–524. [CrossRef]
56. Pandit, H.J. Data Privacy Vocabulary (DPV) Version 1, 2022. Final Community Group Report 05 December 2022. Available online: https://w3c.github.io/dpv/dpv/ (accessed on 5 December 2022).
57. Zhou, J.; Gollman, D. A fair non-repudiation protocol. In Proceedings of the 1996 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 6–8 May 1996; pp. 55–61. [CrossRef]
58. Akaichi, I.; Kirrane, S. A Semantic Policy Language for Usage Control. In Proceedings of the Poster and Demo Track and Workshop Track of the 18th International Conference on Semantic Systems co-located with 18th International Conference on Semantic Systems (SEMANTiCS 2022), Vienna, Austria, 13–15 September 2022; Volume 3235.
59. Robaldo, L.; Batsakis, S.; Calegari, R.; Calimeri, F.; Fujita, M.; Governatori, G.; Morelli, M.; Pacenza, F.; Pisano, G.; Satoh, K.; et al. Compliance checking on first-order knowledge with conflicting and compensatory norms—A comparison among currently available technologies. *Artif. Intell. Law* **2023**, *in press*. [CrossRef]
60. Pandit, H.J. Making Sense of Solid for Data Governance and GDPR. *Information* **2023**, *14*, 114. [CrossRef]
61. van Dijck, J.; Jacobs, B. Electronic identity services as sociotechnical and political-economic constructs. *New Media Soc.* **2020**, *22*, 896–914. [CrossRef]
62. Braun, C.H.J.; Papanchev, V.; Käfer, T. SISSI: An Architecture for Semantic Interoperable Self-Sovereign Identity-Based Access Control on the Web. In Proceedings of the ACM Web Conference 2023, Austin, TX, USA, 30 April–4 May 2023; pp. 3011–3021. [CrossRef]
63. Liu, Y.; Lu, Q.; Paik, H.Y.; Xu, X.; Chen, S.; Zhu, L. Design pattern as a service for blockchain-based self-sovereign identity. *IEEE Softw.* **2020**, *37*, 30–36. [CrossRef]
64. Nanda, R.; Di Caro, L.; Boella, G.; Konstantinov, H.; Tyankov, T.; Traykov, D.; Hristov, H.; Costamagna, F.; Humphreys, L.; Robaldo, L.; et al. A unifying similarity measure for automated identification of national implementations of European union directives. In Proceedings of the 16th Edition of the International Conference on Artificial Intelligence and Law, London, UK, 12–16 June 2017. [CrossRef]
65. O'Connor, T.; Snyder, P. Self-Review Questionnaire: Security and Privacy, 2021. W3C Group Note. Available online: https://www.w3.org/TR/security-privacy-questionnaire/ (accessed on 16 December 2021).
66. Esposito, C.; Hartig, O.; Horne, R.; Sun, C. Assessing the Solid Protocol in Relation to Security & Privacy Obligations. *arXiv* **2022**, arXiv:2210.08270.