# Foundational Components for B2B Data Sharing Using the Solid Protocol

Andreas Both[1,2,*], Thorsten Kastner[1], Dustin Yeboah[1], Christoph Braun[3], Daniel Schraudner[4], Sebastian Schmid[4], Tobias Käfer[3] and Andreas Harth[4]

[1]*DATEV eG, Nuremberg, Germany,*
[2]*Leipzig University of Applied Sciences, Leipzig, Germany*
[3]*Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany*
[4]*Friedrich-Alexander-Universität Erlangen-Nürnberg, Nuremberg, Germany*
*E-mail: andreas.both@datev.de; andreas.both@htwk-leipzig.de; thorsten.kastner@datev.de; dustin.yeboah@datev.de; braun@kit.edu; daniel.schraudner@fau.de; sebastian.schmid@fau.de; tobias.kaefer@kit.edu; andreas.harth@fau.de*
[*]*Corresponding Author*

Received 19 December 2024; Accepted 26 March 2025

## Abstract

This article introduces foundational components for decentralized B2B data sharing based on the solid protocol, emphasizing data sovereignty, security, and interoperability.

These components are: (1) Authorization app (AuthApp) – facilitating granular control and compliance in access granting and revocation processes; (2) rights delegation proxy (RDP) – supporting controlled delegation of rights, enabling natural persons to act on behalf of organizations while ensuring privacy and traceability; (3) data provisioning proxy (DPP) – allowing seamless and secure data provisioning across organizations while masking the identity of upstream data sources to protect business interests.

*Journal of Web Engineering, Vol. 24_4,* 593–634.
doi: 10.13052/jwe1540-9589.2445

The components enable the creation of end-to-end, standards-based, flexible data value chains. We validate their applicability through a real-world financial services use case involving loan processing, which illustrates data sharing and protection challenges in B2B ecosystems.

**Keywords:** Solid, authorization, access control, data sharing, access granting, zero trust, data sovereignty.

## 1 Introduction

In today's connected data-driven world, data sharing between enterprises and among organizations is commonplace, as both providing and consuming organizations benefit from sharing data in their collaborations. However, the data-sharing processes are not satisfactory, as today's businesses often rely on (1) centralized platforms (such as cloud infrastructures), or (2) ad-hoc solutions (such as email attachments). The disadvantage of centralized platforms is that they require both parties in exchange to use – and thereby trust – the platform. However, the platform may not satisfy all parties' requirements regarding the protection of trade secrets (e.g., who (personal data) trades with whom (metadata about the transaction) about what (the actual data)), and has an incentive to build lock-in effects. In summary, centralized platforms impede **sovereignty**. The disadvantage of ad-hoc solutions is that while they may work spontaneously, the consideration of **security** depends on the skills of the sender, data is often duplicated and thus copies lose **freshness**, and the considerations of **organizational structures** and **legal/contractual obligations** require additional ad-hoc solutions, which can be hard to automate and verify (e.g., sharing with an organization, verifying the authority to share data, fulfilling GDPR [35]). In summary, ad-hoc solutions lack **standards-based** approaches, which implement **generic** functionality, to comply with even basic requirements.

Such requirements of data sovereignty, security, and standards-based solutions [16] become more pressing. For example, endeavors such as the International Data Spaces (IDS),[1] GAIA-X,[2] or Solid Dataspaces (SDS) [22] are gaining momentum.

Building on the SDS approach, we present a concept for sovereign data sharing along a chain of enterprises (i.e., a data value chain [6], cf. [1, 20]).

[1] https://github.com/International-Data-Spaces-Association/IDS-RAM_4_0

[2] https://gaia-x.eu/

We address the challenges highlighted in bold and identify *foundational reusable components* that provide the functionality required for establishing data value chains. As we build on the SDS approach, our concept is built on top of the open web technologies from the Solid protocol family and platform [21, 30, 32, 36, 37], which provide the foundation for our decentralized approach regarding interoperable and sovereign data sharing via web interfaces (cf. [40]).

In our solution, data and the identities are stored at the data-providing party (or a provider of their choice), thus addressing sovereignty. The Solid protocols build on encrypted communication, thus addressing security. Data stays at the source when being shared, thus addressing freshness. Data-sharing purposes can be made explicit, and organization-internal rights can be modeled, thus addressing organizational structures and legal/contractual obligations. With the basis of established web technologies, we provide reusable components, thus addressing standards and generality.

To prove the applicability of our concept and the derived foundational components, we implement a use case from the financial services domain, the processing of a loan request, where an enterprise requests a loan from a bank and is required to provide data made available by the enterprise's tax advisor. This use-case represents most of the requirements, as the data sharing between organizations is represented, as well as the need for hiding data providers along the data value chain. Here, only the directly communicating actors know each other, so the origin of the data passed on must remain hidden while the data is still processed along the business chain.

In this article, we highlight the foundational components of our solution. We designed our components as pure Solid apps, s.t., the compatibility with existing approaches can be guaranteed. In detail, these foundational components are:

- an *authorization app (AuthApp)* [5][3] that manifests data management with integrated support for legal constraints (e.g., GDPR, contractual requirements)
- a *rights delegation proxy (RDP)* [31] to handle the actions of natural persons on behalf of legal entities (organizations) to increase operational compatibility
- a *data provisioning proxy (DPP)* [6] to hide data providers in data value chains from clients to protect the understandable protection needs of organizations.

[3]The corresponding publication received the best paper award at the International Conference on Web Engineering 2024.

Hence, another contribution is the demonstration of a Solid-based system for sovereign B2B data sharing, as we describe the corresponding concepts and their implementation as well as validate the applicability in an integrated demonstrator.

The article is structured as follows. The technical terminology and foundations are described in Section 2 followed by a discussion of the related work in Section 3. In Section 4, we make the case for B2B data value chains, using the particular example of loan requests. In Section 5, we give the requirements for our three foundational components, next to the functionality they implement. The validation of our approach is presented in Section 6, followed by a description of the demonstrator in Section 7. Last, we discuss the three components in Section 8, and conclude in Section 9.

## 2 Preliminaries

The Solid project[4] aims to foster data sovereignty on the web [21, 30, 32, 37], specifically that people using the web should be in control of where their data is stored and who can access it. Such effective data ownership may lead to more informed data sharing on how data is intended to be used while simultaneously improving data privacy.

On the technical side, this is achieved by decoupling agent identity, data storage, and provisioning, and consuming applications (cf. Figure 1). The three elements are connected via open standards, and instances thereof can be dynamically exchanged and replaced on demand. This core feature promises to enable many use cases in web-based data-driven ecosystems.

The *Solid Protocol* [11] by the W3C Solid Community Group[5] is a bundle of specifications defining the behavior of a RESTful [15] web server, the *Solid Pod* (personal online datastore). A Solid Pod is a storage space in which data is stored and made available on the web under access control. In the original spirit of the web and unlike today's centralized cloud services, Solid Pods can be hosted in a decentralized fashion. Data is not necessarily stored in a single location but can be distributed across different Pods. The owners of a Solid Pod have full control over their data. They can determine who has access to their information and which applications are allowed to access it, e.g., to read, write, or append data on their Pods.

[4] cf. https://solidproject.org/

[5] cf. https://www.w3.org/community/solid/

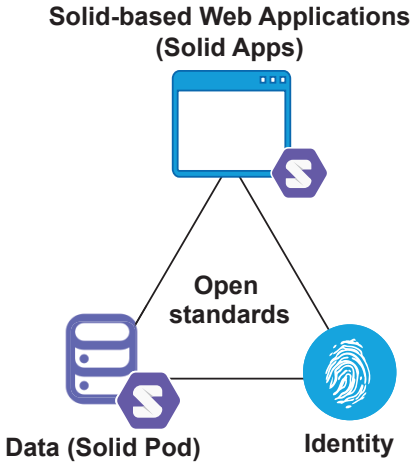


**Figure 1** The Solid principle – decoupling data, application and identity.

The Solid Protocol specifically covers agent identification, authentication, authorization, and data interaction. Agents can be persons, social entities (e.g., groups or organizations), or software. For agent identification, the Solid Protocol relies on WebIDs [29]. A WebID is an HTTP URI that identifies an agent, e.g., users or organizations. Dereferencing an agent's WebID yields the agent's profile document, which contains information about the agent, e.g., the agent's identity provider. For agent authentication, the Solid Protocol relies on Solid-OIDC [13], a modified version of OpenID Connect [28]. Agents authenticate to an OIDC identity provider (IDP) of their choice, e.g., using username and password, and in turn receive a token using which the agent can authenticate themselves at a relying party, e.g., a Web server or service. The agent's identity provider of choice is discoverable from their WebID profile. For agent authorization, the Solid Protocol relies on specifying access control rules following the Web Access Control (WAC) specification [10] or Access Control Policies (ACP) [4]. Using these access control rules, once an agent is authenticated at a Web server or service, the server or service will determine if the agent is allowed to proceed with a certain action on data, e.g., to read or write. Finally, for operations on and management of web resources, Solid borrows heavily from the Linked Data Platform (LDP) [34], effectively extending LDP with access control mechanisms. A web server adhering to the Solid Protocol, i.e., a Solid Pod, thus provides a standardized interface for applications to interact with Pod-stored data. This provides interoperability across applications on an interface level.

If applications know how to process retrieved data from a Solid Pod, data can then be even reused across applications. The Solid Application Interoperability specification [3][6], `INTEROP`, details how applications and agents in general can share and interoperate over data in Solid Pods. To this end, the Resource Description Framework (RDF) provides interoperability through shared vocabularies and data formats, and *Shape Trees*[7] [25] allow for defining schemas to validate the combination of RDF triples. Shape Trees use the capabilities of, e.g., Shape Expressions (ShEx) [26], for the LDP-inspired resource management in Solid Pods. The resource organization in a Pod can be clearly described using the RDF and Shapes Trees, providing a higher level of abstraction. This also provides a potential mechanism for data discovery, allowing Shape Trees to guide applications and users by determining where data can be written to and read from. Conversely, when requesting access to data, Shape Trees allow a data requester to accurately describe the resources they wish to access in a Pod. By comparing the Shape Trees given in an access request with the Shape Trees in the Pod, the relevant resources can be accurately identified, and access grants can be set accordingly (if the data provider wishes to do so). In this way, interoperability on an interface level and on a data level is ensured such that data can be accessed across Solid Pods and then reused across consuming applications.

To be recognized as a Solid-compatible web application[8] (short: *Solid app*), the Solid community has defined a list of inclusion and exclusion criteria. In principle, any web application that complies with the following guidelines is Solid compatible:

- Users must be able to log in using their WebID and refer to the Identity Provider of their choice if identification is required at all.
- Data consumed and generated by an app should be fetched from and stored in one or more Solid Pods.

## 3 Related Work

The Solid Protocol has matured over many years; however, research in the field has only recently gained momentum.

[6] Editor's Draft, 7 November 2023, https://solid.github.io/data-interoperability-panel/specification/

[7] cf. https://shapetrees.org/

[8] cf. https://solidproject.org/apps

Several publications focus on the technical foundations of Solid. For example, in [27], the decentralized verification of data with confidentiality was addressed using Blockchain technology. In [12] and [9], the communications/notifications in the Solid ecosystem are considered. In the context of the data sharing process, [2] dedicated their work to the understandability and usability of the data sharing process for end-users. On the related frontier of data governance, the Data Privacy Vocabulary (DPV) [24] is an ontology for privacy policies, consents, personal data, and other privacy aspects. All definitions of the ontology's terms are derived from the GDPR's definitions. Venturing beyond existing data governance solutions, [42] explores a future of semi-automated data governance at Web-scale, using policy languages to describe data terms of use, and having browsers act on behalf of users to enact policy-based controls. [33] propose a trust-aware framework, TrADS, to integrate data from different Solid Pods in a trust-aware fashion, i.e., determining which data is trustworthy enough to use despite its high heterogeneity. Other research focused on particular use cases: In [38], machine-to-machine sales contracts are considered where Solid Pods are used for data storage. In [41], the data environment for building information modeling (BIM) using Solid is presented. In [8], self-verifying web resource representations using Solid, RDF-Star, and Signed URIs.

B2B data-sharing ecosystems are also envisioned by the communities working on *dataspaces*. Originally, dataspaces should aid in managing multiple different data sources with multiple schemas [16]. In this first vision, single stakeholders lack control and ownership of the different data sources. The International Data Space (IDS) [14, 18, 23] or GAIA-X [7] are (not so) recent initiatives to build data sharing ecosystems focussing on data exchange, governance models, and data sovereignty. However, we find scientific publications to be limited, whereas white papers are broadly available on virtually all issues addressed by the corresponding project's specifications. These specifications, white papers, and other project publications are available on the projects' websites.[9,10]

Addressing the gap between dataspaces and the Solid protocol, [22] presents a conceptual synthesis of Solid as a foundational framework for decentralized dataspaces. These so-called Solid dataspaces build on the Solid protocol to implement the technical details that the IDS protocol mandates.

[9] https://internationaldataspaces.org/publications/most-important-documents/
[10] https://gaia-x.eu/news-publications/publications/

# 4 Ad-hoc B2B Data Value Chains (and Data Value Networks)

In this section, we describe the requirements of B2B data value chains (or more general data value networks). We use a business loan use case for illustration purposes.

## 4.1 Loan Use Case

The loan use case is a business-to-business (B2B) process involving three legal entities – a bank, an enterprise, and a tax advisory office. The latter is a contracted data processor of the enterprise, i.e., the enterprise outsources particular tasks to it. For each legal entity, a natural person is acting on behalf of the legal entity.

Figure 2 outlines our example B2B scenario. Tom, an employee of the (small or medium) enterprise (SME), is mandated to obtain a business loan for the company (i.e., he is acting on behalf of the SME). As a first step, he searches for suitable offers for business loans at the marketplace, with which Lisa has shared her loan offers on behalf of BigBank. Having found an offer with suitable loan terms, he asks BigBank for a specific binding loan offer on behalf of the company. Lisa, an employee of BigBank, is responsible for this task. However, in order to make the SME a concrete loan offer, Lisa needs a business assessment report from the company to determine its financial situation and asks BigBank to share this data. Unknown to BigBank, the SME in turn has already commissioned the tax advisory Office (TAO) to prepare such reports at regular intervals and share them with the SME. Hence, the SME mandated the TAO to prepare the reports and share them with the

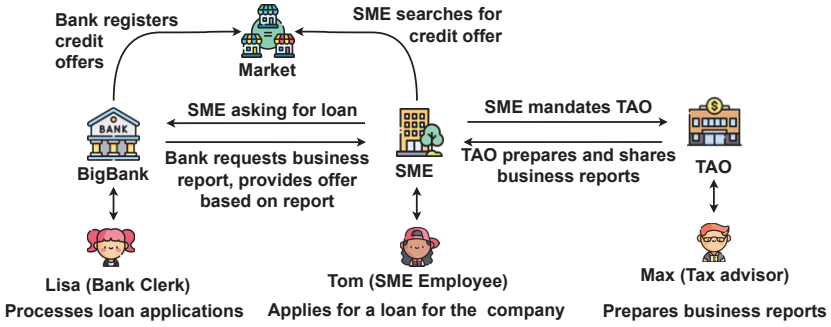


**Figure 2** Outline of the use case.

SME. Max, a tax advisor from the TAO is responsible for this task. Given the required reports are available, Tom then reshares them on behalf of the SME with BigBank, i.e., the data passed on by Max on to SME on behalf of TAO is shared now with the BigBank. Once BigBank has received the data, Lisa can create a concrete and binding loan offer to the SME, s.t., Tom can decide whether to sign up.

### 4.2 B2B Data Value Chain Requirements

Our exemplary use case clearly depicts two crucial factors relevant to initiating and realizing business partnerships. Firstly, to be able to quickly and easily find the required services and the corresponding service providers using a marketplace functionality, and also to be able to easily offer services to potential new customers. Secondly, to enable the secure exchange and forwarding of data/documents required for the processing of a business relationship between several business partners involved.

#### 4.2.1 Requirements for Granting Access

Granting access to entities in a B2B scenario will always be connected to an (explicit or implicit) purpose. Such a purpose might be required due to GDPR (i.e., personal data was shared) or contractual requirements (e.g., exclude the sharing with particular partners). Additionally, the purpose has to be specific and documented to enable later (automatic) compliance checks. The data granting needs to be as flexible as possible to match possible scenarios, e.g., granting/revoking access to specific data items (resources) within a Solid Pod.

In the considered B2B use case, sharing data is the basic requirement for all workflow steps (see Figures 2 and 3), i.e., for accessing general offers via the Marketplace, for providing the required data for computing a specific offer, and finally for storing the loan contract.

#### 4.2.2 Requirements for data sharing between organizations

Without loss of generality, we can derive that, in a data value chain, organizations (legal entities) share data. To be more specific, an organization $O_1$ grants access to the organization $O_2$. With this statement, it is implied that $O_1$ does not know the exact natural persons that will be accessing data. For clarification, $O_1$ should never grant access directly to an employee of $O_2$. However, regarding comprehensibility and security, it is highly demanded to keep a direct connection between the legal entities and the acting natural entities to ensure traceability (often legally required) and prevent loss of

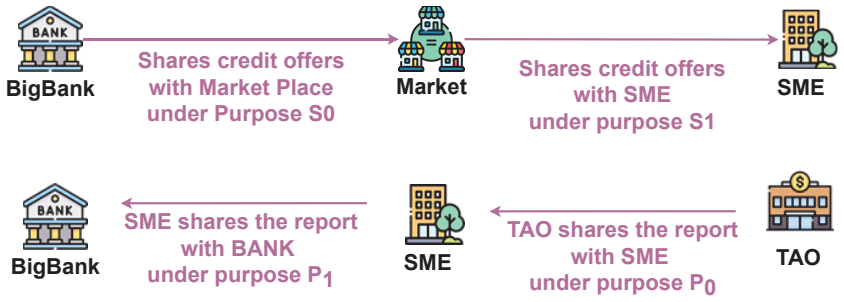


**Figure 3** B2B data sharing chains.

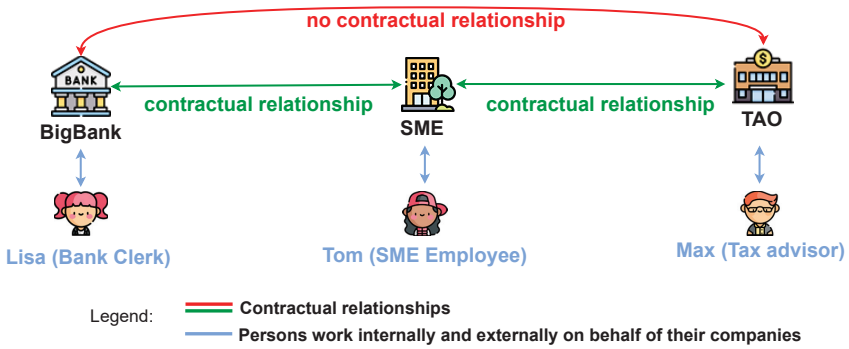


**Figure 4** Involved personas and companies and their contractual and legal relationships to be considered in our use case on the example of business assessment report sharing.

control. In particular, the latter invalidates any solution built on top of a shared account (i.e., the same key is handed to all relevant natural persons). Instead, there is a need for an approach that ensures high-security standards and enables organizations to validate instantly if natural persons are allowed to act on behalf of their organization (i.e., also to revoke earlier granted access rights).

In our use case, the SME is interacting with BigBank, and also with the TAO. Hence, the corresponding natural persons – Lisa (BigBank), Tom (SME), and Max (TAO) – will not interact directly in the described scenario but always on behalf of their organizations (see Figure 4).

### 4.2.3 Data provisioning requirements in B2B data value chains

The basic data sharing in a data value chain is executed directly between two entities. In a B2B scenario, the number of data providers might be very high as it reflects the real-world collaboration of enterprises regarding logistics,

manufacturing, and trade flows (cf. global value chains [1, 39]). However, uncovering the data of all participants of a data value chain has the risk of exposing the data suppliers and finally the whole data value chain. From a business perspective, this is a high risk due to the possibility that a participant collects enough information about its direct and indirect data providers to overtake their operations and therefore threaten these businesses.

In our use case, BigBank should only be aware of the SME while the TAO is completely hidden, i.e., if BigBank requests data from the SME, then it should be handed over (indirectly) from the TAO, s.t., BigBank is not discovering that the TAO is a business partner of the SME. In addition, the functionality of the marketplace should follow the same principle, using a data trustee to manage data access to banks, i.e., the original offer providers are made unrecognizable in the marketplace to protect their business interests.

### 4.3 General Requirement and Summary

There are also general requirements for B2B data value chains in the context of the previous section. In particular, the intended functionality should not create incompatibilities with the Solid ecosystem. Hence, only *pure-Solid components* are considered to fulfill the requirements, i.e., they use the standard interaction and web communication as well as store data only in a Solid Pod.

Finally, we would like to point out a requirement that is often overlooked. Suitable technologies for data sharing must be secure, but also simple and easy to use. As we work on the assumption that company employees must be able to collaborate with a (new) data provider themselves and immediately. For this reason, we refer here to *ad hoc* data value chains.

The requirements are summarized in Table 1.

**Table 1** Requirements for B2B applications

| # | Requirement | Solid |
|---|---|---|
| 1 | Mutual sharing of data | ✓ |
| 2 | Purpose-based data authorizations | ✗ |
| 3 | Authorizations are given for organizations. Natural persons are enabled to act on behalf (i.e., in the name) of an organization. | ✗ |
| 4 | The establishment of data sharing chains with the data supplier must nevertheless protect (i.e., conceal) the business relationships with its data suppliers. | ✗ |
| 5 | Components communicate via well-defined web standards and store data only in Solid Pods | ✓ |

## 5 Concept

After gathering the requirements in relation to sovereign B2B data sharing, we will now present the concepts of our foundational components that are needed for realization. These components include the authorization of data access, the delegation of rights, and indirect data access among several participants.

### 5.1 Authorization

With Solid's core principles to separate the management of identity, data, and applications, users are free to choose applications that fit their specific needs and use these applications with their WebID. As a consequence, different applications may be used to access different types of data with different purposes and needs in mind. While Solid already makes mutual data sharing possible, a data provider in business environments needs to make an informed decision about more complex processes, e.g., across organization boundaries, on what data to share, with whom, and also for which purpose. While data sharing in the form of a proactive grant by the data provider towards another agent is possible, no means to authorize data access requests based on specific purpose demands are specified, including granting or denying them or revoking existing access rights to specific types of data. The first step to do so was made in the W3C Solid Community Group's Application Interoperability Specification.[11]

Sharing data based on a request is an everyday business process independent of specific use cases, so to have reusable components and to enable users to handle data access requests efficiently, we developed a prototype of an application-independent, web-based user interface to authorize access to data in a Solid Pod, the *Authorization App* [5], short AuthApp. The application allows both monitoring and processing of received, existing, rejected, and revoked requests for data sharing. By design, the AuthApp is not integrated directly into the business applications, but exists as a separate service, see also Figure 5.

In our use case, data sharing happens (e.g., between the SME as an agent and BigBank) as another agent, where the SME provides data that contains business reports and BigBank requests access to the business reports. Also, BigBank provides data on a loan offer that the SME requests access to. The

[11] https://solid.github.io/data-interoperability-panel/specification

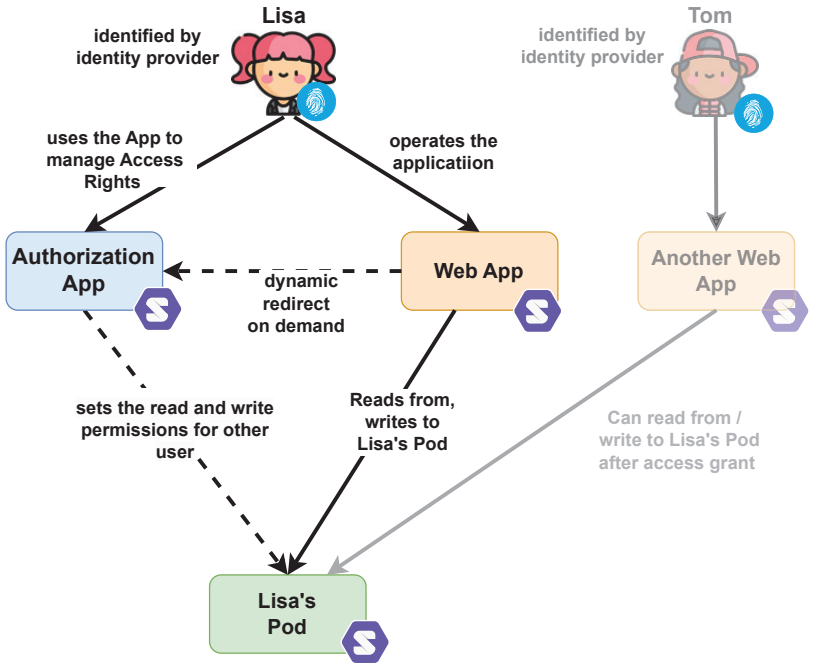


**Figure 5** Component diagram authorization app. Lisa operates her Web App independently of the AuthApp. If Tom wants to access specific data on Lisa's Pod, he sends a data access request to Lisa's Pod, stating additional information like data type and need. Lisa can then decide on and manage Tom's request in the Authorization App.

use cases are different, but the act of sharing data for a specified data need via the AuthApp uses the identical functionality.

The AuthApp for a given agent can be discovered by dereferencing the agent's WebID and extracting the object value of the `interop:hasAuthorizationAgent` statement defined by the Application Interoperability Specification from the RDF graph in the returned identity profile document of the agent. Further, we designed the application to avoid the need to copy or store data outside the personal or company context, meaning all data remains under the user's or company's control. Thus, a business app just needs to redirect the web browser to the corresponding IRI of the AuthApp to provide users with the functionality of managing access to shared data.

The AuthApp gives users the possibility to make fine-grained decisions on different types of data on their pod, based on the actual requests made by others. While the ability to do so is a big improvement in terms of user

empowerment and sovereignty, we see potential pitfalls: on the one hand, users have to decide by evaluating the requesting party's stated needs, which can cause problems for inattentive users that just skim the needs to share data that was not necessary or they did not want to share. On the other hand, the current implementation uses the needs stated by the requesting party. Up to now, there is no way to prevent requests from lying or stating an untrue intention on what data is used for. While such a breach of trust can always be handled afterward by legal means, we see further research needed on how to tighten the control over stated intent and actual usage of data.

### 5.2 Delegation of Rights

Agents may act on behalf of other agents, especially in the business environment, on behalf of organizations, e.g., an employee acts on behalf of their employer. In our use case, Tom, as a natural person, acts on behalf of his employer, the SME, an organization, to sign the loan contract for the SME. The transfer of rights from one agent to another is a so-called delegation or power of attorney. In a delegation, a delegator defines policies that state rights and transactions that may be exercised in the delegator's name towards an affiliate. A delegate may then act towards the affiliate as specified in a related policy. Current solutions for data sharing using Solid are based on, e.g., Access Control Lists[12] (ACL), or if a bigger group is considered on the membership of an agent in vCard groups.[13] Using these solutions in a delegation setting, the delegate's identity and the delegation itself are necessarily revealed to an affiliate who has to set the corresponding ACLs to provide access. When considering the privacy concerning the delegate's potential interest in staying anonymous, issues arise quickly. Additionally, once a delegation is granted, a delegator loses control of whether a delegate uses only legitimate acts within the defined policies. To solve the issues with privacy and control in delegations, we propose the rights delegation proxy (RDP) [31] as an approach for private and legitimate data sharing and delegations. Because delegations occur frequently (especially in business environments along hierarchies [17]), delegations of rights have far-reaching consequences in terms of power for the delegate, and delegation processes are similar across business use cases in terms of the abstract roles of the delegator, delegatee, and affiliate, we created the rights delegation proxy

[12] https://solidproject.org/TR/wac

[13] https://www.w3.org/2006/vcard/ns#Group

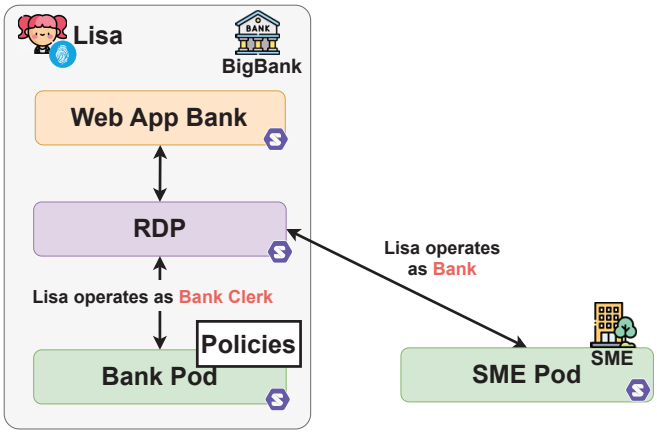


**Figure 6** Component diagram rights delegation proxy (RDP). Lisa uses her role as a clerk in the bank's Web App. When accessing the SME Pod, the RDP checks whether her action is in accordance with BigBank's policies, and if so, enacts her action with the identity of BigBank. The SME will only recognize that BigBank did act.

(RDP), a reusable component that may be used across different organizations (see Figure 6).

Before a delegation starts, we assume that the affiliate has already defined access rights for the delegator, e.g., read rights for the SME on BigBank's Pod. The affiliate does not need to know about potential delegates. To define a delegation from the delegator to the delegatee, the delegator creates a policy that states the delegate's rights, e.g., tied to a specific WebID or to an employee role in an organization. The policy also defines the conditions under which the delegate may act. Conditions are evaluated by the RDP and can define how a resource must look before a delegate may access it (pre-condition), or how the resource must look after the delegate accesses it (post-condition). Note that even though the delegator can, in principle, delegate as many rights to the delegatee as they want, the delegator can never lift the rights the affiliate has admitted to them. For example, if the delegator has read rights, only read rights could be granted to a delegatee, but not for writing, as a right to write would be missing from the affiliate towards the delegator. Policies can be more complex and, e.g., be defined as Shape Expressions (ShEx)[14] or SPARQL[15] ASK queries.

[14]https://shex.io/
[15]https://www.w3.org/TR/sparql11-overview/

An organization delegatee sends HTTP requests directly to the RDP where the access path is the affiliate's resource, and the query contains the host of the affiliate's URI. After receiving a request, the RDP looks up fitting policies depending on the WebID and requested web resource. The RDP checks pre-conditions with a "preflight GET" to the requested resource and evaluates if the response matches the pre-condition. To check a post-condition, the RDP evaluates the message body, which contains the to-be-expected resource state. In any case, the RDP logs the checks' results, including time, content, accessed resource, and requesting WebID at a location defined by the delegator. If the delegate fulfills the policy, the RDP authenticates as the delegator s.t., a separation between the delegatee and the affiliate is made, and forwards the delegatee's request to the affiliate's resource and sends the respective response back to the delegatee. From the affiliate's perspective, only the delegator has been involved.

The RDP prevents actions from happening outside the defined control of the delegator and thus is most interesting for business use cases. As a component in the overall process, the RDP has a centralized role as it has to manage, evaluate, and enact all incoming requests from potential delegatees. Its high responsibility is thus a bottleneck and a potential single point of failure. Additionally, the responsibility to define policies that mirror the delegator's intent is a delicate step that should be handled carefully, as a delegatee may act directly on behalf of the delegator, as is already done today.

### 5.3 Indirect Data Access

While the RDP is aimed towards delegating rights from one agent to another with respect to a known affiliate's resource, the RDP benefits mostly the accessing party, the delegate, protecting its identity from the affiliate. In a related case, the identity of the party whose data is accessed shall be protected and not be disclosed. We propose the data provisioning proxy (DPP) (Figure 7), as presented in [6]. The process is similar to the way the RDP works, cf. Section 5.2, but differs mostly in the way policies and the data flow are handled. Here, the DPP is located at a passing party (e.g., the SME) that receives data requests from an external party (e.g., from Lisa on behalf of BigBank) where the requested data might be retrieved from an external, hidden source. The DPP checks whether the requested resource is located on the current Pod (e.g., for SME's own data), or if the resource needs to be retrieved and passed along from an external data source (e.g., a third party like the TAO that shared access with the SME). Again, the owner of the external data source may define a policy under which conditions sharing data

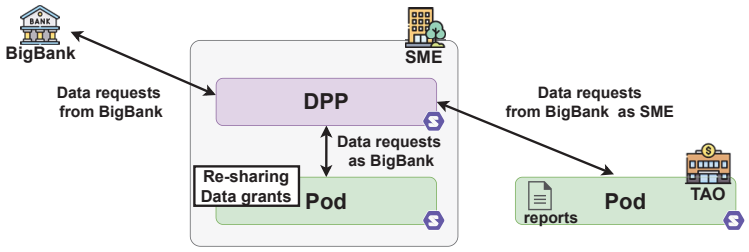


**Figure 7** Component diagram data provisioning proxy (DPP). BigBank wants to access a resource with reports that are believed to be stored on the SME Pod. the SME's DPP forwards the request and retrieves current data from the TAO without disclosing TAO's identity.

is acceptable. The DPP validates the policy with respect to the requested data and, if sharing is allowed, the DPP retrieves the respective resource. Note that the DPP assumes again the identity of the passing party; here, the DPP is authenticated as the SME. From the view of the requesting party, the provided data looks as if it originates from the passing party, while the original data source is masked. While the RDP is mostly concerned with enacting actions and thus needs an efficient evaluation of onsite delegator policies, the DPP needs to focus on efficient data localization to determine if requested data is on the passing party's Pod or an external Pod. An included bonus of the DPP is the possibility for the passing party to process the requested data in a last step before eventually sharing it with the requester. This is especially important if the data to be shared is primarily about the passing party (e.g., the TAO's reports can be anonymized by the SME before sharing it with BigBank). In any case, the passing party represents the data source for such data, while the original source is still under the control of the external party, hidden away.

## 6 Validation

In this section, we validate that the presented components (see Section 5) provide the functionalities required to build B2B data value chains (Section 4). We show how the single components are connected and work in unison to support the B2B data sharing in our example use case.

### 6.1 Advertising (Loan) Contract Offers on the Marketplace

To enable initiating a business loan contract, the bank needs to advertise that it offers business loans to potential business partners. To this end, the bank's

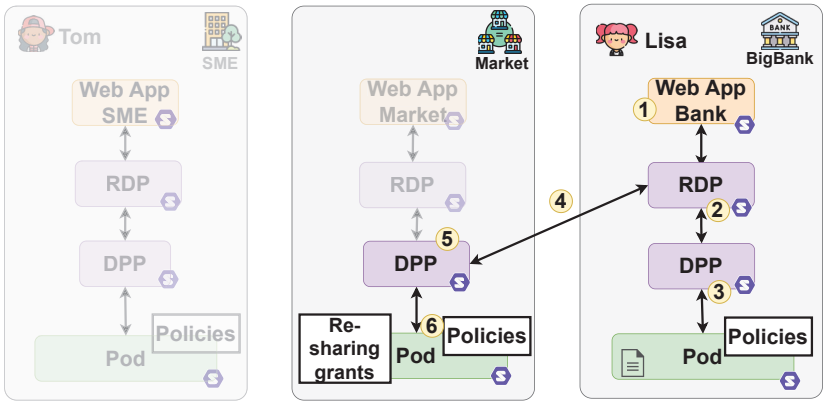


**Figure 8** Loan contract advertisement via a marketplace platform. Components not relevant to the data sharing chain are grayed out.

employee creates an advertisement for loan offers on the bank's Pod and links to that advertisement at the marketplace. The technical interaction sequence is (visualized in Figure 8):

1. Lisa logs into the BigBank business Web App with her own WebID. To create a new advertisement for loan offers on the Pod of BigBank, she sends an authenticated request to the RDP of BigBank.
2. The *BigBankRDP* authenticates Lisa and checks if she is authorized according to BigBank's policies to create new advertisements for loan offers on BigBank's Pod. It then forwards Lisa's request to the *BigBankDPP*.
3. As the resource to be created is an actual Pod-stored resource, the *BigBankDPP* forwards the request to BigBank's Pod. The Pod checks access control rules and creates the advertisement resource.
4. Next, Lisa creates a link to the advertisement for BigBank's Pod on marketplace's Pod. Her request again is checked by the *BigBankRDP*. The RDP then forwards Lisa's request to the *marketplace DPP*, authenticated as BigBank.
5. The *marketplace DPP* receives the authenticated request coming from BigBank to add a link to the new offer to marketplace's catalog. It checks whether the requested resource is an actual Pod-stored resource. Additionally, it checks if BigBank is allowed to add new links to the catalog.
6. The *marketplace DPP* performs an authenticated request as marketplace to marketplace's Pod. The link is added to the catalog.

## 6.2 Data Sharing via a Data Value Chain

In the following sections, we describe how the data is shared along a data value chain. In our use case, we have two instances of such data provisioning: The advertisement for loan offers and the business assessment reports.

### 6.2.1 Sharing of advertisements for loan offers

On the marketplace, potential business partners are able to discover the link to the bank's loan offer. Note that the advertisement resource is stored at the bank and only linked to from the marketplace. What is more, the link under which the advertisement is *accessible through* the marketplace's catalog is not the same link that is *registered at* the catalog by the bank's employee Lisa. This means that the URI of the bank's advertisement is not discoverable from the catalog, but the catalog assigns a new URI to that advertisement, essentially creating a mapping between the two URIs. Upon request, the advertisement is thus requested using the catalog-assigned link, and the request is internally forwarded using the registered original link. The technical interaction sequence is (visualized in Figure 9):

1. Tom logs in to the SME business Web App using his WebID. To retrieve the advertisements for loan offers on the marketplace's Pod, he sends an authenticated request to the RDP of the SME.
2. The *SMERDP* authenticates Tom and checks if he is authorized by the SME's policies to interact with the marketplace.
3. The *marketplace DPP* receives the authenticated request coming from the SME to access the advertisements for loan offers. It checks whether

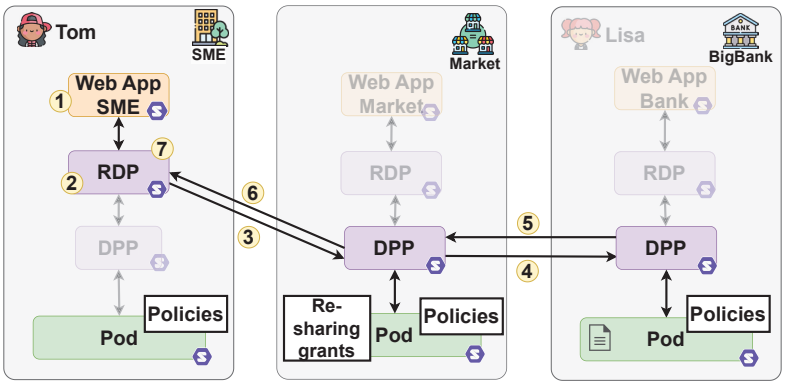


**Figure 9** Data value/sharing chain for advertisements for loan offers. Components not relevant to data sharing chain are grayed out.

the requested resource is an actual Pod-stored resource or an external one to be forwarded. Additionally, it checks if redistribution is allowed.
4. On pass, the *marketplace DPP* performs an authenticated request as the marketplace for BigBank's loan offer ads.
5. The *BigBankDPP* receives an authenticated request from the marketplace to the resource containing the advertisement data. As this resource is an actual Pod-stored resource, the request is forwarded. The Pod checks access control rules and returns the data via the proxy.
6. The *marketplace DPP* receives the response, logs it, and responds to the request of the SME.
7. The *SMERDP* receives the response from the marketplace and forwards it to Tom's Web App.

### 6.2.2 Sharing of business assessment reports

In the subsequent loan initiation process, BigBank requests business assessment reports from the SME to check their financial situation. These business reports, however, are not created by the SME but are provided to them by the TAO. In particular, the reports are stored on the TAO's Pod and are only made available to the SME. The SME is then able to make these reports available virtually to BigBank. Upon request at the SME, data is dynamically fetched from the TAO and forwarded to BigBank without creating copies of data. The technical interaction sequence is (visualized in Figure 10):

1. Lisa logs in to the business Web App using her own WebID. To retrieve the business report data on the SME's Pod, she sends an authenticated request to the RDP of BigBank.

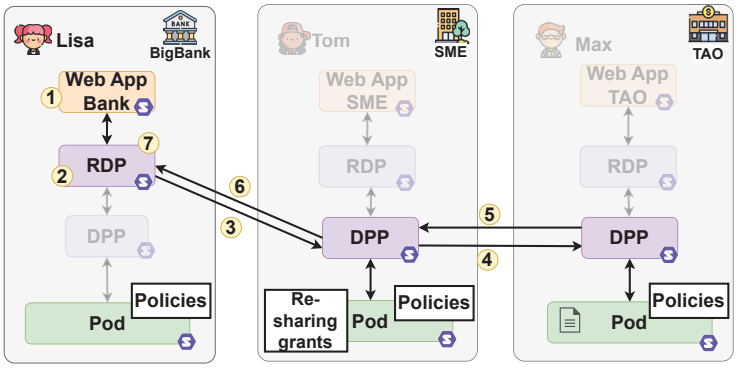


**Figure 10** Data value/sharing chain for business assessment reports. Components not relevant to the data sharing chain are grayed out.

2. *BigBank's RDP* authenticates Lisa and checks if she is authorized by the bank's policies to interact with the SME. Then it requests the resource from the SME, authenticated as BigBank.
3. The *SME's DPP* receives the authenticated request coming from BigBank to access the business report data. It checks whether the requested resource is an actual Pod-stored resource or an external one to be forwarded. It also checks if redistribution is allowed.
4. On pass, the *SME's DPP* performs an authenticated request as the SME for the TAO's data.
5. The *TAO's DPP* receives an authenticated request from the SME to the resource containing the business report data. As this resource is an actual Pod-stored resource, the request is forwarded. The Pod checks access control rules and returns the data via the proxy.
6. The *SME's DPP* receives the response, logs it, and responds to the request of BigBank.
7. The *BigBank's RDP* receives the response from the SME and forwards it to Lisa's Web App.

## 7 Demonstrator

In addition to the theoretical outline of how the components interplay (see Section 6), we present our demonstrator that showcases the implementation of the foundational components and associated applications in our B2B use case. First, we introduce the components that comprise our system architecture. Then, we illustrate our implementation of the three use case phases: Loan advertisement, the loan approval process, and contract termination.

### 7.1 System Architecture

Our demonstrator[16,17] is composed of the following components for each company (see Figure 11):

*Solid Business Web Apps* enable the employees to carry out their business activities. For the example of the SME, the initiation of a loan inquiry, the provisioning of the business reports requested by the bank to prepare a loan offer, and the approval of the Bank's loan offer. In addition, an authorization application, as described in [5], is used to manage incoming, existing, rejected, and revoked requests for data sharing.

[16] https://github.com/mandat-project/hackathon-demo
[17] https://github.com/mandat-project/delegation-proxy

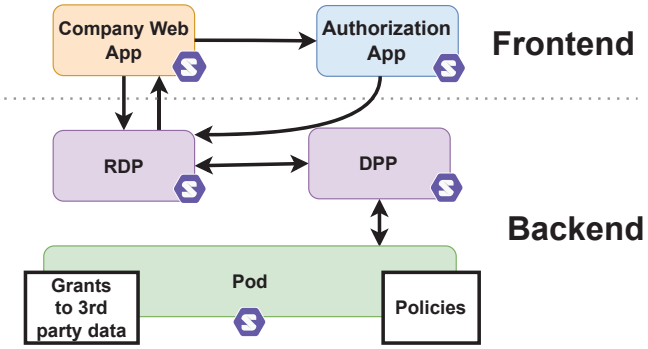


**Figure 11** System components architecture.

A *company's Solid Pod* stores the company's business data and makes it available under access control. Notably, it also contains rights delegation policies, i.e., rules that define which specific employees are allowed to interact internally and externally on behalf of the company. Further, it contains definitions for resources that are made available via proxied data provisioning.

An *Authorization App* (see Section 5.1) [5], enables users to share data based on requests. The AuthApp handles data access requests towards the company's Solid Pod and allows employees to monitor and process received, existing, rejected, and revoked requests for data sharing of company data, based on a defined purpose for data access as defined by GDPR. The AuthApp is a separate service that can run independently of business applications, such that reusability is given, also across different data fiduciaries, organizations, and users.

A *rights delegation proxy (RDP)* (see Section 5.2), introduced in [31], receives and logs requests made by the company's employee (e.g., Tom). It authenticates the employee using their WebID and checks if they were delegated the required rights to proceed with their request. To this end, the RDP retrieves and validates corresponding policies defined by the delegator (e.g., the SME) from their Pod. If all requirements are fulfilled, the RDP proceeds with the delegatee's request, but updates the authentication headers with the delegator's credentials. Any received response is logged and forwarded to the delegatee.

*A data provisioning proxy (DPP) (see Section 5.3)*, as envisioned in [6], receives all data requests (e.g., from Lisa on behalf of BigBank) and checks on the Pod whether the requested resource is an actual Pod-stored resource

or to be retrieved and passed along from an external data source. That is, it validates if the company's own data (the SME) or data from a third party (the TAO shared with the SME) is requested. If the requested data originates from a third party, the DPP checks the sharing policy of the third party. If allowed, the DPP retrieves the resource (authenticated as the SME) and provides it as if originating from the company (the SME), masking the original data source (the TAO).

## 7.2 Our Use Case in Three Phases

All of the aforementioned components are used in the entire process of our demonstrator, which can be divided into three successive phases: In the first phase, illustrated by Figure 12, the marketplace acts as a platform that enables the SME to find a suitable loan provider (BigBank). In the second phase, illustrated by Figure 13, the SME sends a loan request to the previously selected BigBank. However, before BigBank offers a loan, it needs to evaluate the associated risk. To this end, it requires business assessment reports from the SME, which are created by the TAO and shared with BigBank via the SME in a data provisioning chain. After reviewing the business assessment reports, BigBank makes an offer and the SME can now decide to accept it. In the third and final phase, illustrated by Figure 14, once a loan agreement has been concluded, BigBank can terminate the contract. When this happens, the SME can withdraw all authorizations that were previously granted as part of the loan approval process.

### 7.2.1 Marketplace for Discovering Advertised Loan Offerings

In this section, we describe our implementation of Phase 1 from our use case, illustrated in Figure 12. We assume the following initial state in our demonstrator. An advertisement for loan offers is already created at

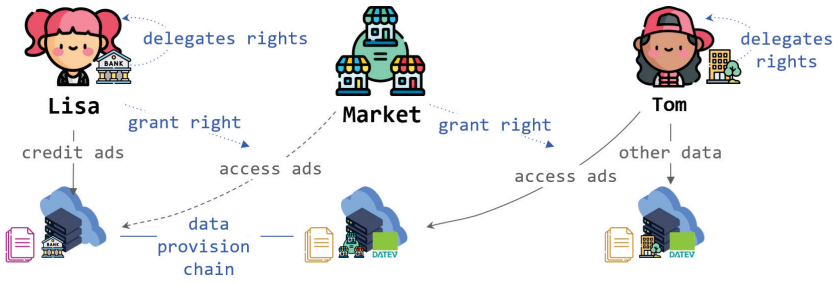


**Figure 12** Phase 1 – discovery of the loan offer advertisement.

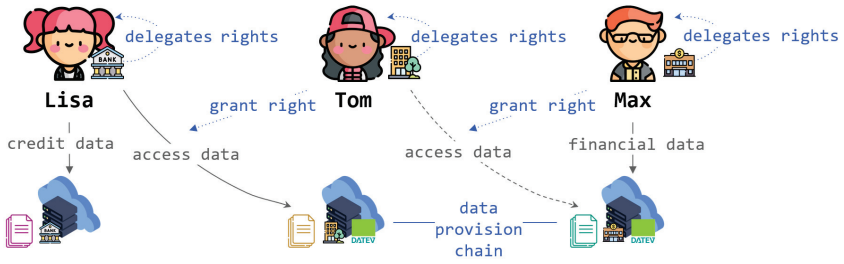


**Figure 13** Phase 2 – loan approval process.

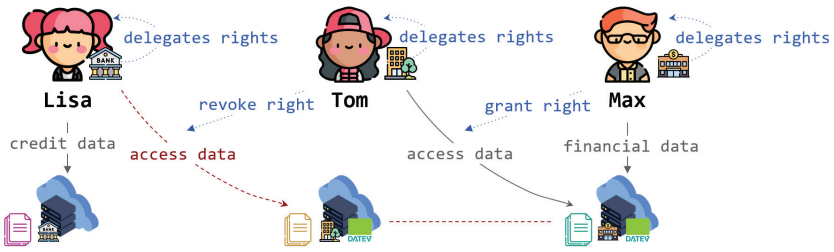


**Figure 14** Phase 3 – termination of an active loan contract.

**Screencast: https://purl.archive.org/mandatb2b/JWE2024**

BigBank's Pod. An associated link at the marketplace's catalog is already registered, and corresponding access rights are in place. That is, the marketplace enables Tom, on behalf of the SME, to find a suitable loan offering service provider. In our demo, the services offered are either consumer loans or business loans. Ideally, a prior process, e.g., as outlined in Section 6.1, would allow service providers, such as BigBank, to register and share their service advertisements.

Our demonstrator starts with Tom, employee of the SME, who wants to apply for a loan on behalf of his employer. To this end, Tom logs in to the business web app that allows him to discover loan services offered on the marketplace. Note that Tom logs in with his employee WebID, see the header bar in Figure 15. In the corresponding WebID profile, it is asserted that Tom is an employee of the SME and that the SME's RDP is to be used for outgoing requests. Thus, Tom will act on behalf of the SME in cases where the SME's RDP allows requests made by Tom in this context.

Using the business web application, Tom thus searches for suitable loan offerings. In the marketplace's catalog, all advertisements are categorized via ShapeTrees and accessible to requesting participants. The catalogue is

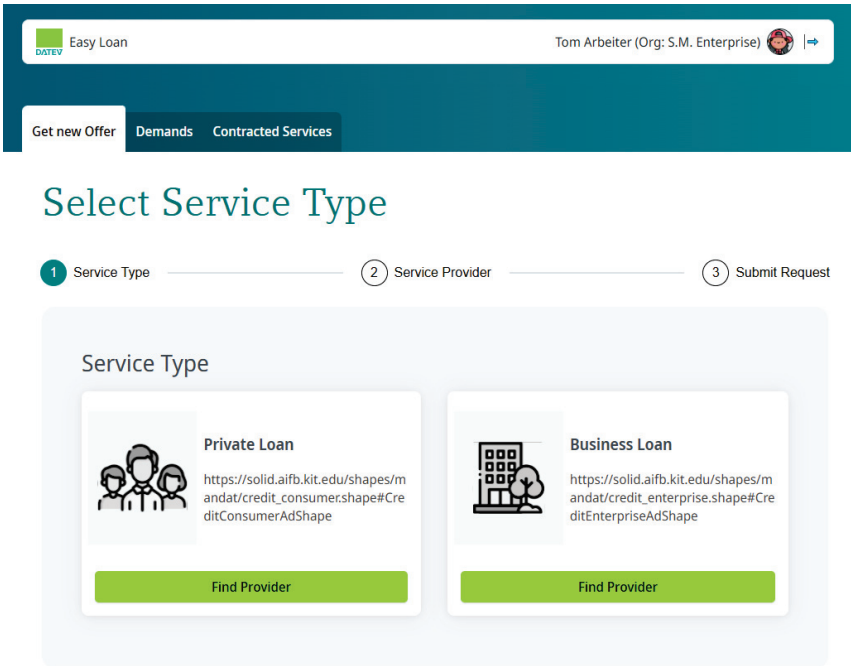


**Figure 15** Tom selects service type in the marketplace on behalf of the SME.

queried for all the different loan service offered. On the market's side, any request is authenticated to be sent by the SME. The fact that Tom initiated the request is not revealed via the RDP.

We highlight in addition that there are no advertisement resources stored at the marketplace's Pod. Only links to the corresponding advertisements are registered at the marketplace. When the marketplace's catalog is queried, the catalog's representation is retrieved. This representation includes the catalog's URIs for loan advertisements, not the original URIs from BigBank's Pod, for example. When particular advertisements are requested, the marketplace dynamically looks up the mapping between the catalog's advertisement URIs and the registered URIs, and retrieves the advertisements via the marketplace's DPP.

This is, for example, the case when Tom selects the service type of business loans (see Figure 15) to then see a particular advertisement for a business loan (see Figure 16). Details of the particular loan offerings are displayed by the business web application. This is an implementation of the process described in Section 6.2.1.

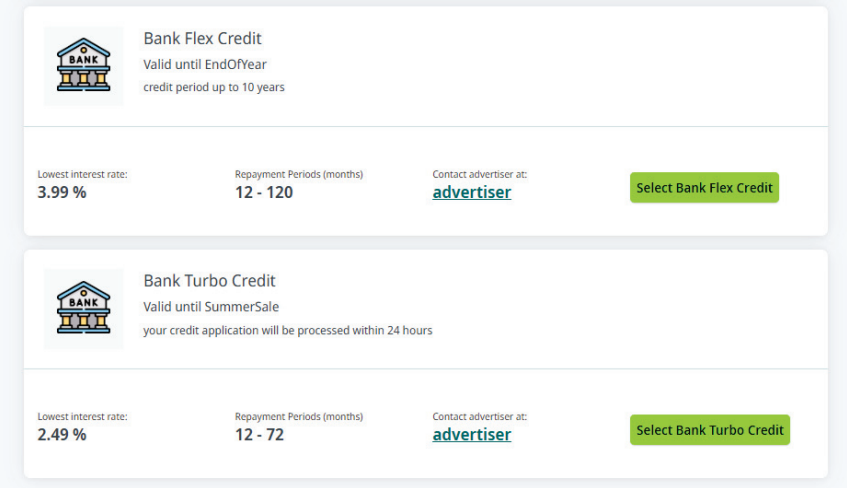


**Figure 16** Tom selects the service provider on behalf of the SME.

### 7.2.2 Loan Approval Process

In this section, we describe our implementation of Phase 2 from our use case, illustrated in Figure 13. There are multiple processes that intertwine in the loan approval process: There is the ordering process which consists of the SME's demand, BigBank's offer and the SME's order of a business loan. In that process, BigBank requires assessing the risk of providing the SME with a loan and requests business reports from the SME who rely on the TAO to provide the requested reports. Across these two data exchange processes, access rights to corresponding resources are requested and granted (or declined) upon request inspection.

By selecting a particular loan offering advertisement (see Figure 16), Tom starts the loan approval process on behalf of the SME. A loan demand is issued to BigBank's Pod. Again, the RDP checks and takes over Tom's request. Loan demands are only readable by BigBank that will only recognize that the SME requested a loan.

Lisa (representing BigBank) uses BigBank's business web application to handle loan requests. Lisa is logged in to that web application similar to Tom; Lisa is acting on behalf of BigBank via BigBank's RDP. To review the SME's loan demand, she requires access to the SME 's business assessment reports for risk evaluation. To obtain this data, Lisa sends an access request to the SME's Pod asking for the business assessment reports. As Lisa's request is sent on behalf of BigBank via BigBank's RDP, the access request asks the

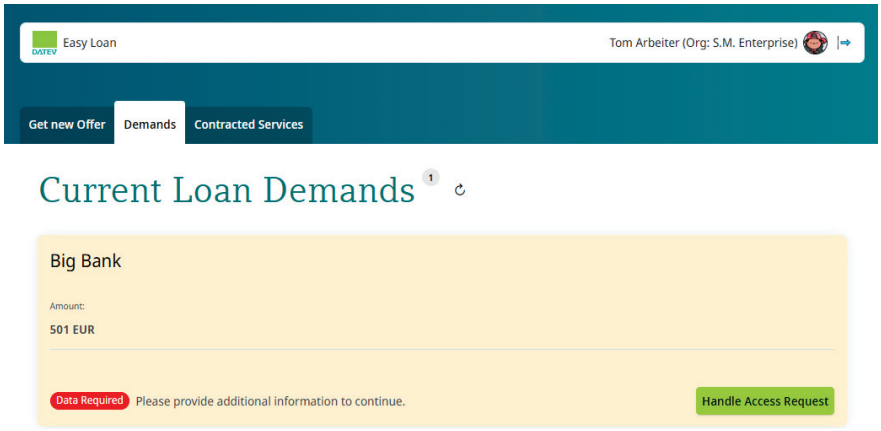


**Figure 17** Tom on behalf of the SME is notified about a data access request from BigBank in the SME's business app.

SME (and not Tom) to grant read access to BigBank (and not Lisa) on the business reports. Tom is notified about the existence of the access request in the business web application. A red badge indicates a pending access request concerning the loan demand, i.e., the business process (see Figure 17). The business application is not processing the access request. To handle the access request, Tom is redirected to the *Authorization App*.

Tom is again logged in to the AuthApp on behalf of the SME. Therefore, Tom is able to handle access requests to the SME's Pod if the policies defined by the SME's RDP allow Tom to do so. Tom inspects the access request, where all necessary information is detailed: BigBank requests read access on SME's business reports for the purpose of fulfilling contractual obligations (see Figure 18). Tom then authorizes the access for BigBank (not Lisa), and the AuthApp sets the corresponding access rules in the SME's Pod and logs authorization and data grants for later compliance audits.

Back on BigBank's side, Lisa can now access and review the existing business assessment reports of the SME (see Figure 19). Lisa has only access to this data as she is acting on behalf of BigBank through the RDP. After analyzing the reports, Lisa realizes that more up-to-date data is required. Lisa then issues a document request to the SME's Pod.

Tom receives the document request through his business application. Tom then forwards the document request to the TAO's Pod, instructing them to provide more recent data. As usual, the document request is officially attributed to and issued by the organization through the acting employee.

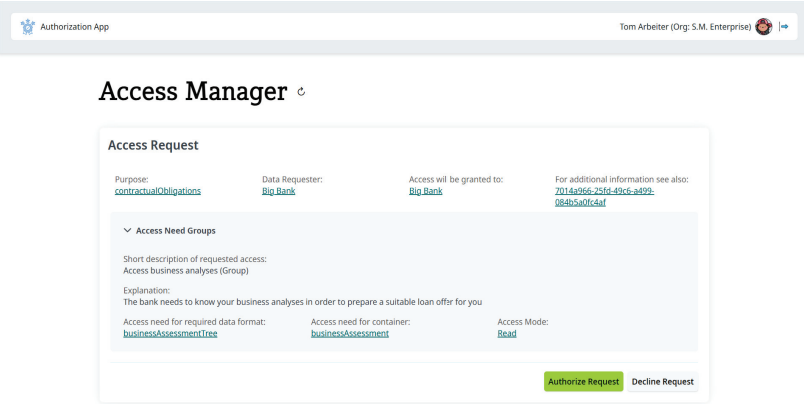


**Figure 18** Tom on behalf of the SME handles the access request for business assessment data from BigBank in the AuthApp.

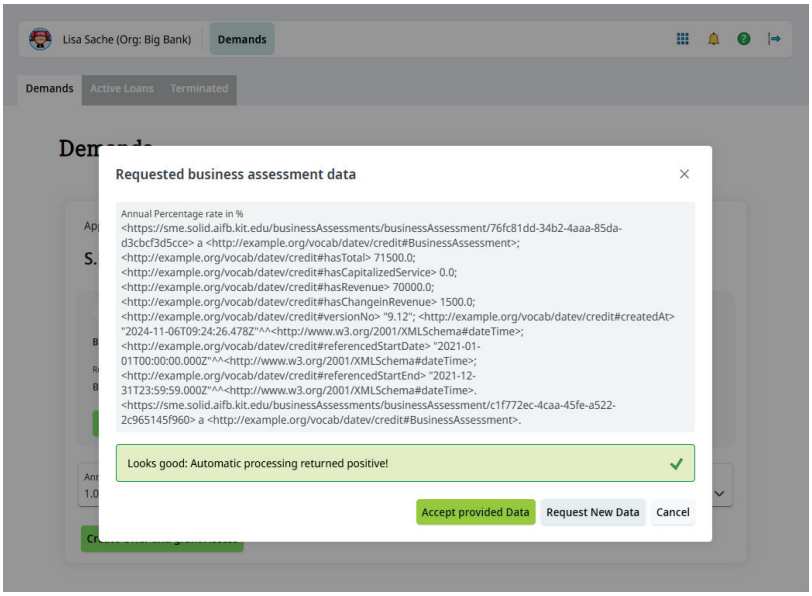


**Figure 19** Lisa acting on behalf of BigBank inspects the business assessment data in the BigBank app.

Max, who works for the TAO, handles the document request in their business application, similarly logged in and using the TAO's RDP as the other employees. Tom prepares the new reports and stores them in the TAO's Pod. The new reports are accessible for the SME as the TAO set up the access

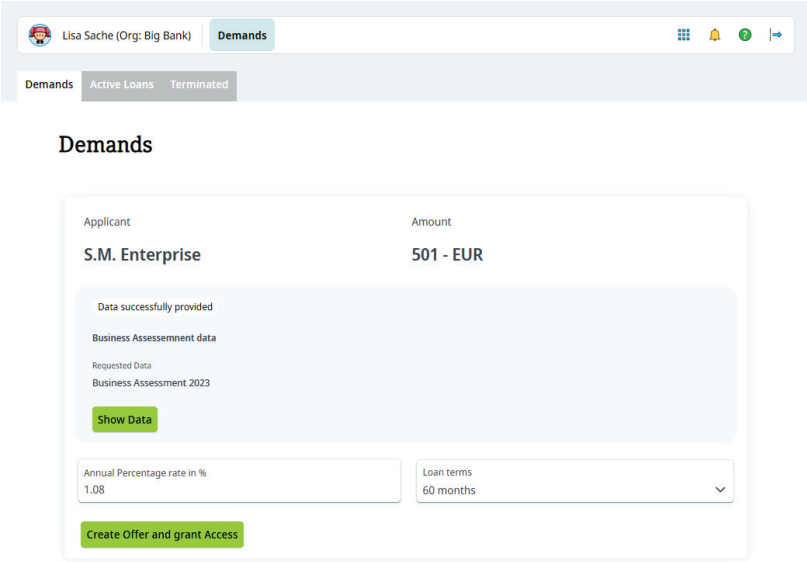


**Figure 20** Lisa creates on behalf of BigBank a loan offer for the SME in the BigBank app.

control that way. Because the SME has access to the new business reports and because the SME authorized BigBank to view their business reports, the SME's DPP is configured to also provide the new reports to BigBank when requested. Thus, a data provisioning chain is put in place, as outlined in Section 6.2.2.

Lisa now has all the necessary data and creates a loan offer for the SME in the BigBank Pod (see Figure 20). In order for Tom to see this offer, BigBank must grant read access to the SME. In addition, the SME needs to be authorized to issue an order for a loan to BigBank's Pod. To this end, Lisa is redirected to the AuthApp and handles this self-issued access request. Again, the AuthApp sets the desired access rules and logs authorizations for compliance. Tom can now inspect the offer and accept it by creating a loan order in BigBank's Pod (see Figure 21).

### 7.2.3 Contract Termination

In this section, we describe our implementation of Phase 3 from our use case, illustrated in Figure 14. For active loans, BigBank has the option of terminating them (see Figure 22), e.g., when the term comes to an end or contractual conditions and obligations are not met. As in the processes before, Lisa acts on behalf of BigBank in this matter.

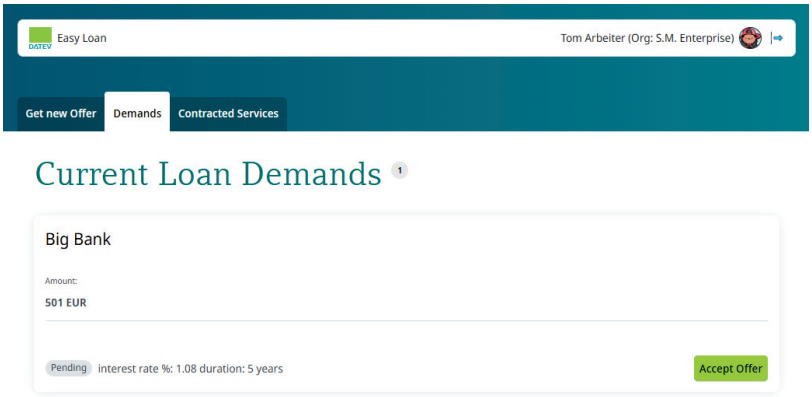


**Figure 21** Tom reviews and accepts the loan offer within the SME app on behalf of the SME.

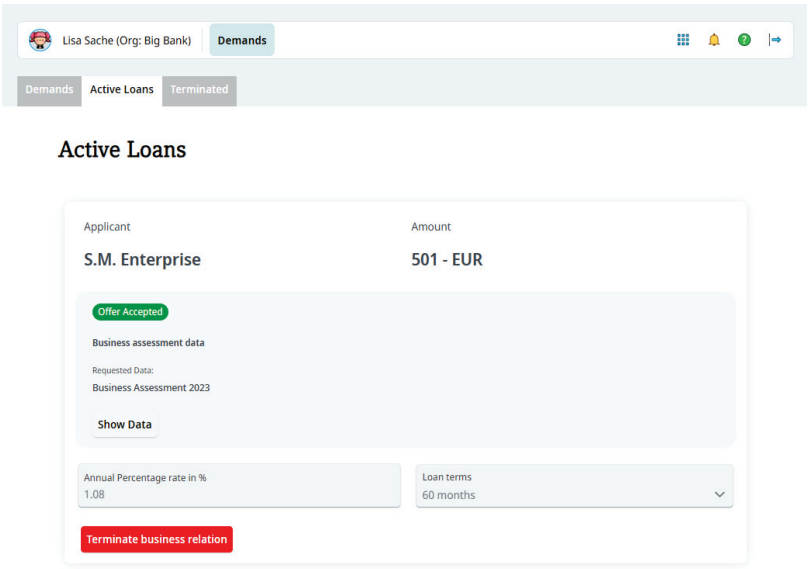


**Figure 22** Lisa terminates the active loan contract on behalf of BigBank.

After termination, the loan then appears as terminated also for Tom in his business application (see Figure 23). As the business relation is terminated, Tom has the option to revoke the data access authorizations that grant BigBank access to business reports. After a redirect to the AuthApp, Tom can now remove access authorizations on behalf of the SME. Partial or all access

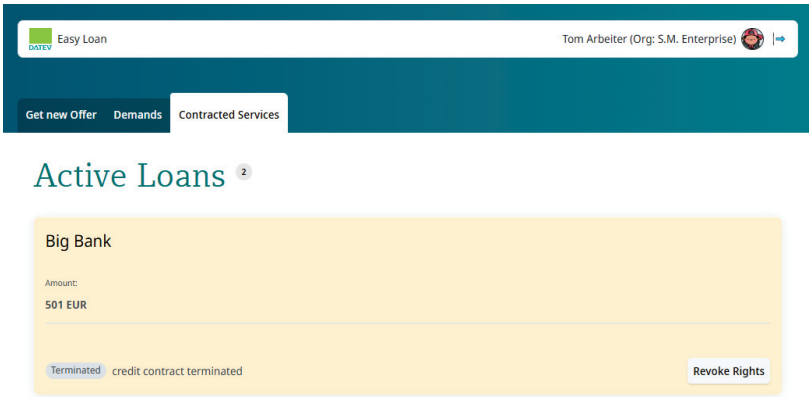


**Figure 23** Tom views the terminated loan on behalf of the SME.

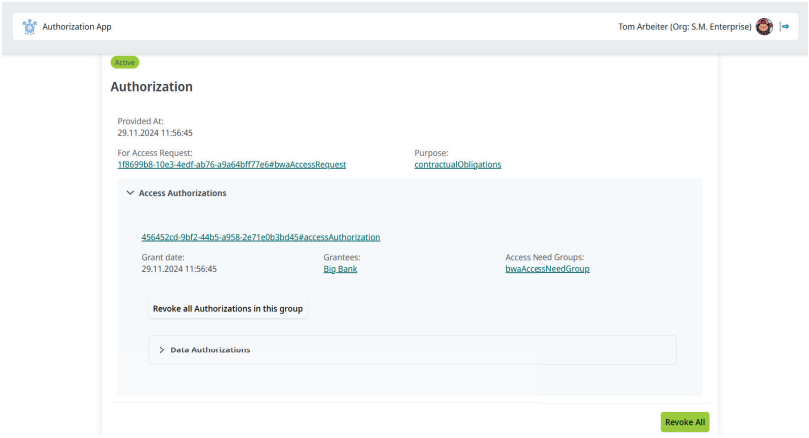


**Figure 24** Tom revokes authorizations in the *Authorization App* on behalf of the SME.

rights are revoked that were previously granted in the loan approval process (see Figure 24).

At this point, regulation may require the SME to keep certain access authorizations for compliance on BigBank's side. In any case, authorizations relating to the particular business process are to be frozen. Only strictly required data and no newly created data may be accessible by BigBank in the context of this business loan. There may exist other business relations that require access to some of the resources. In this case, these other authorizations need to be taken into account. This is a feature we plan to support in a future version of the AuthApp.

## 8 Discussion

In this section, we discuss the presented foundational components to enable sovereign end-to-end data sharing in B2B data-driven ecosystems. In general, our implementation shows that creating a Solid-based ecosystem is possible and thus illustrates the required foundational components. This section is an overview of the considerations from [5], [6] and [31].

**For authorization** (see Section 5.1), we covered the *Authorization App* [5]. Although our approach is applicable and represents a cornerstone for such ecosystems, further research is needed to achieve the goal of seamless initiation of data sharing. That is, the reasoning on instances of purposes ontology needs to be safe and sound to establish completely automatable processes. In addition, we still use an Inbox mechanism, which has since been removed from the `INTEROP` specification to establish an initial connection between the prospective data processor and the data provider. While this enables a safe and sound way to initialize a data chain using just a data structure definition (`shapetrees`), it lacks a GDPR-conform representation of the purpose of the data request. However, the inbox mechanism follows the GDPR principle of using only data necessary in relation to the purposes (data minimization) and avoidance of keeping data for no longer than is necessary (storage limitation), as it does not share any information about the Pod of the future data provider. Additionally, the `INTEROP` specification does not specify how to deal with withdrawn or outdated authorizations. Actually, it is not defined how to represent a withdrawn access authorization (and how it differs from a valid one). An authorization app can only recognize that a data access grant is outdated by the fact that a new data release description is created with the reference that this replaces the old one via the property `interop:replaces`. Hence, the withdrawn permit does not contain any indication that it has been replaced by a new one. Therefore, extensions of the `INTEROP` specification (or an additional specification) are required to overcome these challenges, which might otherwise render the functionality unusable.

Although the RDF is well-suited for representing data in a machine-interpretable form, it needs to be validated if it is acceptable that the stored data about the granted and denied requests (including requested data formats, requesters, and beneficiaries) can be changed by the user. From a legal perspective, there might be a need for a write-only data container, s.t., this data stays consistent and is immutable. Furthermore, a standardization of the vocabulary for describing the structure of Solid Pods seems to be reasonable to prevent implementation-specific behavior that might reduce the portability.

Finally, as useful and convenient as an external authorization app is for the user, the issue might be raised that users expect that this is the only way they handle data requests to their Pod. However, applications might still internally create the `ACL` without redirecting to the authorization app. This could imply different, inconsistent, or even non-GDPR-conforming data representations, which might become problematic in a well-regulated business context and therefore hinder the adoption of Solid technologies. From this observation, one might derive the need to extend the Solid protocol, s.t., users or organizations can restrict the authorization apps that are allowed to change such data in their Solid Pods.

**For delegation of rights** (see Section 5.2), we covered the RDP [31] as a facilitator between a delegator, the respective delegate, and an affiliate. All delegated actions pass through the RDP, which authenticates as the delegator. Thus, the RDP is a core component in the B2B context. The power over any actions to be taken is shifted to the delegator who has exclusive control over policies. At the same time, responsibility is shifted away from the affiliate who merely interacts with the delegator. The delegatee's potential actions are defined by the delegator's policies. The RDP solves the problem of preserving the privacy of the delegatee: Authenticated as a delegator, there is no difference in the action's origins towards an affiliate, while the delegatee's identity is not revealed. Additionally, the RDP increases the delegator's control and traceability over the delegatee's actions.

The RDP is currently a centralized component that manages all incoming delegated requests. This poses a potential bottleneck and single point of failure. However, requests are independent of each other; multiple instances of an RDP may run in parallel behind a load balancer.

Policy implementation is dependent on the use case. Complex, custom policies, e.g., in large organizations, are supported: pre- and post-conditions can be defined for the expected shape of data, e.g., using SPARQL queries or ShEx/SHACL shapes. Only if the policy's conditions are met is the delegatee's request forwarded.

Organizations often use Business Process Model and Notation (BPMN) to describe more complex workflows, e.g., a contract may only be signed after an accountant agrees. Ontologies like WiLD [19] can be similarly used to represent and monitor workflows. More importantly, during the execution of such semantic workflows, the RDP checks conditions with a privacy-respecting and secure delegation mechanism.

**For indirect data access** (see Section 5.3), we covered the data provisioning proxy [6] which enables data value chains. Data provisioning is facilitated

by a dynamic data lookup that ensures hiding the original data source. This may lead to scalability and availability issues.

Consider the basic data exchange between two parties, the data consumer and the data provider, who is also the original data source. Compare this to a data value chain scenario where the data provider is dynamically fetching data from another data provider, who in turn fetches the data, and so on. Data retrieval may take only a very short time in the basic case, but with a growing number of provisioning chain participants, request latency leads to growing runtime. In addition, instead of relying on only one data source to be available in the basic case, with more participants, the risk of an unavailable data source that breaks the data provisioning chain also grows.

Thus, scalability (in terms of request latency) and availability (in terms of technical outages) pose major practical challenges. One potential solution strategy may be creating a dedicated data fiduciary that provides guarantees towards the availability and scalability of their data provisioning service.

## 9 Conclusions and Future Work

In this article, we presented foundational components for decentralized standards-based sovereign sharing of data along a B2B data value chain (data value network). The introduced pure-Solid components are listed below. They are derived from the requirements of organizations in a data-driven (i.e., data-sharing) economy:

- The *Authorization App* (AuthApp) to manage and process (grant, revoke) access requests to web resources.
- The *Rights Delegation Proxy* (RDP) to allow agents (here: natural persons) to act on behalf of other agents (here: legal entities, e.g., organizations).
- The *Data Provisioning Proxy* (DPP) to provision data retrieved from upstream data sources, without disclosing the source (e.g., a second-level data provider).

We validated the applicability of those components in a standard process that involves data sharing between businesses: finding business partners, the request for a loan, and its approval.

However, these components are not tied to our use case or implementation. They form reusable components for other data ecosystems where authorizations need to be managed, delegation needs to be implemented, and data provisioning from third-party sources is required. For the first time,

we established pure-Solid components dedicated to users with very high demands regarding security and traceability. Due to the characteristics of Solid apps, they are also reusable (i.e., it is possible to integrate them into custom processes). Due to their conception, we suggest that the AuthApp and the RDP and DPP address the basic needs of each enterprise; therefore, we consider them *foundational Solid components*

With our contributions, we continue to work on our long-term agenda of providing standard functionality for data-driven ecosystems as Solid apps. Hence, we envision a future environment where Solid apps for business can be implemented rapidly while still meeting the highest demands in relation to data sovereignty, data protection, interoperability, security, traceability, and the protection of (trade) secrets. In addition, such ecosystems are becoming even more important in the age of AI, as data/networks form the central basis for specific AI models, but still need to be shared responsibly in B2B data spaces so as not to jeopardize business success.

The foundational components presented in this article provide significant advancements in enabling B2B data sharing using the Solid Protocol. However, several areas require further exploration to enhance the robustness, security, and applicability of these solutions.

*Authorization and data control:* Preventing users from unintentionally sharing data (in contradiction to legal or contractual requirements) remains a challenge due to the complexity and the situational parameters. Extending policies, as utilized in the RDP and AuthApp, to encompass specific data types could be a promising direction. Research into enforcing control over the stated intent and actual data use is essential, potentially leveraging trusted execution environments, as envisioned in the International Data Spaces (IDS) framework. Additionally, the creation of an immutable protocol or data container to store configurations and histories of data sharing would provide greater legal safety. From a technical standpoint, integrating reasoning capabilities into the AuthApp application is necessary, as current mechanisms lack semantic validation of data formats such as `shapetrees`.

*Data provisioning:* For scenarios like data provisioning, mechanisms ensuring that intermediaries cannot alter data beyond the stated intent are crucial to prevent fraud. The policies defined in the RDP and DPP should be refined and extended to support complex business scenarios, thereby increasing trust and utility in real-world applications.

*Scalability and automation:* While our approach addresses many core challenges, achieving fully automatable and scalable processes requires further advancements. For instance, ensuring safe and sound reasoning over

instances of purpose ontologies is critical. Additionally, the reliance on redirect chains via mechanisms to obscure the original source of data introduces scalability and availability concerns. Implementing a data trustee could mitigate these issues.

*Towards a global B2B data ecosystem:* Our vision is to expand this solution into a global ecosystem of interconnected Solid Pods, enabling businesses to thrive through the collaborative use of semantic Linked Data. This goal requires additional research to address technical and organizational blockers, including traceability, business secret protection, and compliance with data sovereignty principles.

These future directions aim to enhance the practicality and adoption of our foundational components, paving the way for robust and secure B2B data-driven ecosystems.

## Acknowledgments

This work has been supported in part by the German Ministry for Education and Research (BMBF) under grant 16DTM107 (*MANDAT*), specifically FKZn 16DTM107A, 16DTM107B, and 16DTM107C, which are funded by the European Union – NextGenerationEU.

## References

[1] Pol Antràs and Davin Chor. Chapter 5 – global value chains. In Gita Gopinath, Elhanan Helpman, and Kenneth Rogoff, editors, *Handbook of International Economics: International Trade, Volume 5*, volume 5 of *Handbook of International Economics*, pages 297–376. Elsevier, 2022.

[2] Hadrien Bailly, Anoop Papanna, and Rob Brennan. Prototyping an end-user user interface for the Solid Application Interoperability Specification under GDPR. In Catia Pesquita, Ernesto Jimenez-Ruiz, Jamie McCusker, Daniel Faria, Mauro Dragoni, Anastasia Dimou, Raphael Troncy, and Sven Hertling, editors, *The Semantic Web*, pages 557–573. Springer Nature, 2023.

[3] Justin Bingham, Eric Prud'hommeaux, and Elf Pavlik. Solid Application Interoperability. W3C Editor's Draft., 2023.

[4] Matthieu Bosquet. Access control policy (acp). Editor's draft, W3C Solid Community Group, 2022.

[5] Andreas Both, Thorsten Kastner, Dustin Yeboah, Christoph Braun, Daniel Schraudner, Sebastian Schmid, Tobias Käfer, and Andreas Harth. AuthApp – portable, reusable solid app for GDPR-compliant access granting. In *Web Engineering: 24th International Conference, ICWE 2024, Tampere, Finland, Proceedings*, page 199–214. Springer-Verlag, 2024.

[6] Andreas Both, Dustin Yeboah, Thorsten Kastner, Daniel Schraudner, Sebastian Schmid, Christoph Braun, Andreas Harth, and Tobias Käfer. Towards Solid-based B2B data value chains. In *21st Extended Semantic Web Conference (ESWC 2024)*, 2024.

[7] Arnaud Braud, Gaël Fromentoux, Benoit Radier, and Olivier Le Grand. The road to European digital sovereignty with GAIA-X and IDSA. *IEEE Netw.*, 35(2):4–5, 2021.

[8] Christoph H.-J. Braun and Tobias Käfer. Self-verifying web resource representations using Solid, RDF-star and signed URIs. In *The Semantic Web: ESWC 2022 Satellite Events*, pages 138–142, Cham, 2022. Springer.

[9] Christoph H.-J. Braun and Tobias Käfer. Web push notifications from Solid Pods. In Tommaso Di Noia, In-Young Ko, Markus Schedl, and Carmelo Ardito, editors, *Web Engineering*, pages 487–490, Cham, 2022. Springer International Publishing.

[10] Sarven Capadisli. Web access control. Editor's draft, W3C Solid Community Group, 2022.

[11] Sarven Capadisli, Tim Berners-Lee, Ruben Verborgh, and Kjetil Kjernsmo. Solid Protocol, December 2021.

[12] Sarven Capadisli, Amy Guy, Christoph Lange, Sören Auer, Andrei Sambra, and Tim Berners-Lee. Linked data notifications: A resource-centric communication protocol. In *The Semantic Web*, pages 537–553, Cham, 2017. Springer International Publishing.

[13] Aaron Coburn, Elf Pavlik, and Dmitri Zagidulin. Solid-OIDC, March 2022. https://solidproject.org/TR/oidc.

[14] Edward Curry, Simon Scerri, and Tuomo Tuikka. *Data Spaces: Design, Deployment, and Future Directions*, pages 1–17. Springer International Publishing, Cham, 2022.

[15] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[16] Michael J. Franklin, Alon Y. Halevy, and David Maier. From databases to dataspaces: a new abstraction for information management. *SIGMOD Rec.*, 34(4):27–33, 2005.

[17] Michele M Hughes. Remedying financial abuse by agents under a power of attorneys for finances. *Elder's Advisor*, 2:39, 2000.

[18] Matthias Jarke, Boris Otto, and Sudha Ram. Data sovereignty and data space ecosystems. *Business & Information Systems Engineering*, 61:549–550, 2019.

[19] Tobias Käfer and Andreas Harth. Specifying, monitoring, and executing workflows in linked data environments. In *The Semantic Web – ISWC 2018 – 17th International Semantic Web Conference, Monterey, CA, USA, Proceedings, Part I*, volume 11136 of *Lecture Notes in Computer Science*, pages 424–440. Springer, 2018.

[20] Liena Kano, Eric W. K. Tsang, and Henry Wai-chung Yeung. Global value chains: A review of the multi-disciplinary literature. *J. of International Business Studies*, 51(4), 2020.

[21] Essam Mansour, Andrei Vlad Sambra, Sandro Hawke, Maged Zereba, Sarven Capadisli, Abdurrahman Ghanem, Ashraf Aboulnaga, and Tim Berners-Lee. A demonstration of the Solid platform for social web applications. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, page 223–226. International World Wide Web Conferences Steering Committee, 2016.

[22] Sascha Meckler, Rene Dorsch, Daniel Henselmann, and Andreas Harth. The Web and Linked Data as a Solid Foundation for Dataspaces. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, page 1440–1446. Association for Computing Machinery, 2023.

[23] Boris Otto, Michael ten Hompel, and Stefan Wrobel, editors. *Designing Data Spaces: The Ecosystem Approach to Competitive Advantage*. Springer, 2022.

[24] Axel Polleres, Beatriz Esteves, and Bert Bos. Data Privacy Vocabulary (DPV). Final Community Group Report, Data Privacy Vocabularies and Controls Community Group, May 2022.

[25] Eric Prud'hommeaux and Justin Bingham. Shape Trees Specification. https://shapetrees.org/TR/specification/.

[26] Eric Prud'hommeaux, José Emilio Labra Gayo, and Harold R. Solbrig. Shape expressions: an RDF validation and transformation language. In Harald Sack, Agata Filipowska, Jens Lehmann, and Sebastian Hellmann, editors, *Proceedings of the 10th International Conference on*

*Semantic Systems, SEMANTiCS 2014, Leipzig, Germany, September 4-5, 2014*, pages 32–40. ACM, 2014.

[27] Manoharan Ramachandran, Niaz Chowdhury, Allan Third, John Domingue, Kevin Quick, and Michelle Bachler. Towards complete decentralised verification of data with confidentiality: Different ways to connect Solid Pods and blockchain. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 645–649. Association for Computing Machinery, 2020.

[28] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore. OpenID connect core 1.0. Final specification, 2014.

[29] Andrei Sambra, Henry Story, and Tim Berners-Lee. Webid 1.0 – web identity and discovery. W3c editor's draft, W3C, 2014.

[30] Andrei Vlad Sambra, Essam Mansour, Sandro Hawke, Maged Zereba, Nicola Greco, Abdurrahman Ghanem, Dmitri Zagidulin, Ashraf Aboulnaga, and Tim Berners-Lee. Solid: a platform for decentralized social applications based on linked data. *MIT CSAIL & Qatar Computing Research Institute, Tech. Rep.*, 2016.

[31] Sebastian Schmid, Daniel Schraudner, and Andreas Harth. The Rights Delegation Proxy: An Approach for Delegations in the Solid Dataspace. In *Proceedings of the Second International Workshop on Semantics in Dataspaces (SDS 2024) co-located with the 21st Extended Semantic Web Conference (ESWC 2024)*, 2024.

[32] Oshani Seneviratne, Amy van der Hiel, and Lalana Kagal. *Tim Berners-Lee's Research at the Decentralized Information Group at MIT*, page 201–213. ACM, 1 edition, 2023.

[33] Valentin Siegert, Dirk Leichsenring, and Martin Gaedke. Trusting decentralized web data in a solid-based social network. In Kostas Stefanidis, Kari Systä, Maristella Matera, Sebastian Heil, Haridimos Kondylakis, and Elisa Quintarelli, editors, *Web Engineering – 24th International Conference, ICWE 2024, Tampere, Finland, June 17-20, 2024, Proceedings*, volume 14629 of *Lecture Notes in Computer Science*, pages 230–245. Springer, 2024.

[34] Steve Speicher, John Arwe, and Ashok Malhotra. Linked Data Platform 1.0. W3c recommendation, W3C, 2015.

[35] The European Parliament and the Council of the European Union. Regulation (EU) 2016/679 (General Data Protection Regulation) GDPR, 2016.

[36] Tim Berners-Lee et al. Solid.

[37] Ruben Verborgh. *Re-decentralizing the Web, For Good This Time*, page 215–230. ACM, 1 edition, 2023.

[38] Xinni Wang, Christoph H.-J. Braun, Andreas Both, and Tobias Käfer. Using schema.org and Solid for linked data-based machine-to-machine sales contract conclusion. In *Companion Proceedings of the Web Conference 2022*, WWW '22, page 269–272. Association for Computing Machinery, 2022.

[39] Zhi Wang, Shang-Jin Wei, Xinding Yu, and Kunfu Zhu. Characterizing global value chains: Production length and upstreamness. Working Paper 23261, National Bureau of Economic Research, March 2017.

[40] Jeroen Werbrouck, Pieter Pauwels, Jakob Beetz, and Léon van Berlo. Towards a decentralised common data environment using linked building data and the Solid ecosystem. In *36th CIB W78 Conference*, pages 113–123, 2019.

[41] Werbrouck, Jeroen and Pauwels, Pieter and Beetz, Jakob and van Berlo, Léon. Towards a decentralised common data environment using linked building data and the solid ecosystem. In *Advances in ICT in Design, Construction and Management in Architecture, Engineering, Construction and Operations (AECO) : Proceedings of the 36th CIB W78 2019 Conference*, pages 113–123, 2019.

[42] Jesse Wright, Beatriz Esteves, and Rui Zhao. Me want cookie! towards automated and transparent data governance on the web. *CoRR*, abs/2408.09071, 2024.

## Biographies

**Andreas Both** is Head of Research at DATEV eG (a top-tier business software provider in Germany) and a professor at the Leipzig University of Applied Sciences (Germany) where he leads the Web & Software Engineering (WSE) research group which focuses on safe, secure, decentralized, and data-driven architectures as well as applied AI.

**Thorsten Kastner** works as a software consultant engineer at DATEV eG, Nuremberg. He received his doctoral degree at the Technical Faculty of the Friedrich-Alexander-Universität Erlangen-Nürnberg. His fields of interest are Semantic Web technologies, machine learning, and digital signal processing.

**Dustin Yeboah** works as a software developer at DATEV eG, Nuremberg. He focuses on general web development with a specialization in Semantic Web technologies and data ecosystems.

**Christoph Braun** is a doctoral researcher in the Web Science group at Karlsruhe Institute of Technology (KIT). He focuses his research on methods to build semantic data-sharing ecosystems based on existing and emerging Web standards.

**Daniel Schraudner** is currently working towards a Ph.D. degree with the Chair of Technical Information Systems at Friedrich-Alexander-Universität Erlangen-Nürnberg. His research interests include Solid, Linked Data, as well as Semantic Web technologies and knowledge representation in general. He received his B.Sc. (2016) and M.Sc. (2019) degrees in computer science from Friedrich-Alexander-Universität Erlangen-Nürnberg.

**Sebastian Schmid** received his Master's degree in Computational Engineering from Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany, in 2019. He is doing his Ph.D. studies at FAU on the topic of distributed networked data and agent-based systems. Research focuses on self-adaptation and control of agents in dynamic environments, especially in situations where little or no information is available for decision-making.

**Tobias Käfer** leads the Web Science group at Karlsruhe Institute of Technology (KIT) in Germany as an interim professor. His research interests are in decentralized and distributed knowledge graph-based AI systems.

**Andreas Harth** holds the Chair of Technical Information Systems at the Friedrich-Alexander-Universität Erlangen-Nürnberg in Germany. His research interests are large-scale data interoperation on the Semantic Web, Linked Data, knowledge representation, computational logic, and user interaction on web data.