ONTO-DESIDE GA NUMBER: 101058682

ONTO-DESIDE

DELIVERABLE

Digital twin concept design, including ontology-based data sharing platform architecture and methodology – Report v1

Deliverable number	D4.1
Deliverable name	Digital twin concept design, including ontology-based data sharing platform architecture and methodology
Work package	WP4
Lead partner	IMEC
Contributing partners	LIU, CIRC, FAS
Deadline	2023-02-28
Dissemination level	Public
Date	2023-02-28

Funded by the European Union

PROJECT INFORMATION

Project summary

Circular economy aims at reducing value loss and avoiding waste, by circulating materials or product parts before they become waste. Today, lack of support for sharing data in a secure, quality assured, and automated way is one of the main obstacles that industry actors point to when creating new circular value networks. Together with using different terminologies and not having explicit definitions of the concepts that appear in data, this makes it very difficult to create new ecosystems of actors in Europe today. This project will address the core challenges of making decentralised data and information understandable and usable for humans as well as machines. The project will leverage open standards for semantic data interoperability in establishing a shared vocabulary (ontology network) for data documentation, as well as a decentralised digital platform that enables collaboration in a secure and privacy-preserving manner.

The project addresses several open research problems, including the development of ontologies that need to model a wide range of different materials and products, not only providing vertical interoperability but also horizontal interoperability, for cross-industry value networks. As well as transdisciplinary research on methods to find, analyse, and assess new circular value chain configurations opened by considering resource, information, value, and energy flows as an integral part of the same complex system. Three industry use cases, from radically different industry domains, act as drivers for the research and development activities, as well as test beds and demonstrators for the cross-industry applicability of the results. The developed solutions will allow for automation of planning, management, and execution of circular value networks, at a European scale, and beyond. The project thereby supports acceleration of the digital and green transitions, automating the discovery and formation of new collaborations in the circular economy.

Project start date and duration

1st of June 2022, 36 months

Project consortium

No	Partner	Abbreviation	Country
1	Linköping University	LIU	Sweden
2	Interuniversitair Micro-Electronica Centrum	IMEC	Belgium
3	Concular Ug Haftungsbeschrankt	CON	Germany
4	+Impakt Luxembourg Sarl	POS	Luxembourg
5	Circularise Bv	CIRC	The Netherlands
6	Universitaet Hamburg	UHAM	Germany
7	Circular.Fashion Ug (Haftungsbeschrankt)	FAS	Germany
8	Lindner Group Kg	LIN	Germany
9	Ragn-Sells Recycling Ab	RS	Sweden
10	Texon Italia Srl	TEXON	Italy
11	Rare Earths Industry Association	REIA	Belgium

Document reference

Project acronym	Onto-DESIDE			
Programme	Horizon Europe			
Grant agreement number	101058682			
Project URL	https://ontodeside.eu/			
EU Project Officer	Giuseppina LAURITANO			
Project Coordinator	Name	Eva Blomqvist	Phone	+46 13 28 27 72
	E-mail	eva.blomqvist@liu.se	Phone	
Project Manager	Name	Svjetlana Stekovic	Phone	+46 13 28 69 55
	E-mail	svjetlana.stekovic@liu.se	Phone	+46 701 91 66 76
Deputy PC	Name	Olaf Hartig	Phone	+46 13 28 56 39
	E-mail	olaf.hartig@liu.se	Phone	
Deliverable name	Digital twin concept design, including ontology-based data sharing platform architecture and methodology – Report v1			
Deliverable number	D4.1			
Deliverable version	V1.0			
Deliverable nature	Report			
Dissemination level	Public			
Due date	M9			
Delivery date	28/02/2023			
Keywords	Digital twin, Ontology, Open Circularity platform, Architecture, Methodology			

Document change log

Version	Date	Description	Authors	Checked by
0.1	14.2.2023	First version	Gertjan De Mulder, Martin Vanbrabant, Ben De Meester	Olaf Hartig
0.2	27.2.2023	Incorporating feedback	Gertjan De Mulder, Ben De Meester	Mikael Lindecrantz
1.0	28.2.2023	Incorporating feedback	Gertjan De Mulder, Ben De Meester	

Document approval

Version	Date	Name	Role in the project	Beneficiary

Table of Content

1 Abbreviations

Abbreviation	Explanation
ACP	Access Control Policy
API	Application Programming Interface
CSS	Community Solid Server
FAIR	Findable, Accessible, Interoperable, Reusable
LDP	Linked Data Platform
OIDC	OpenID-Connect
R2RML	Relational to RDF Mapping Language
RDF	Resource Description Framework
RML	RDF Mapping Language
TRL	Technology Readiness Level
WP	Work Package
YAML	Yet Another Markup Language
YARRRML	Yet Another R2RML/RML Mapping Language

2 Introduction

Semantic interoperability of data is one of the biggest barriers towards data sharing in the Circular Economy. Onto-DESIDE aims to provide the technical foundations for information flows that will transform European Industry towards a Circular Economy, by means of digitalisation and data sharing. The project leverages a decentralised digital platform that enables collaboration in a secure manner. This allows for automation of discovery, planning, management, and execution of cross-industry circular value networks, at a European scale and beyond. Combined with access control policies for data privacy and confidentiality, automation is enabled whilst protecting company-internal data, and allows data sharing to happen at the right level of granularity.

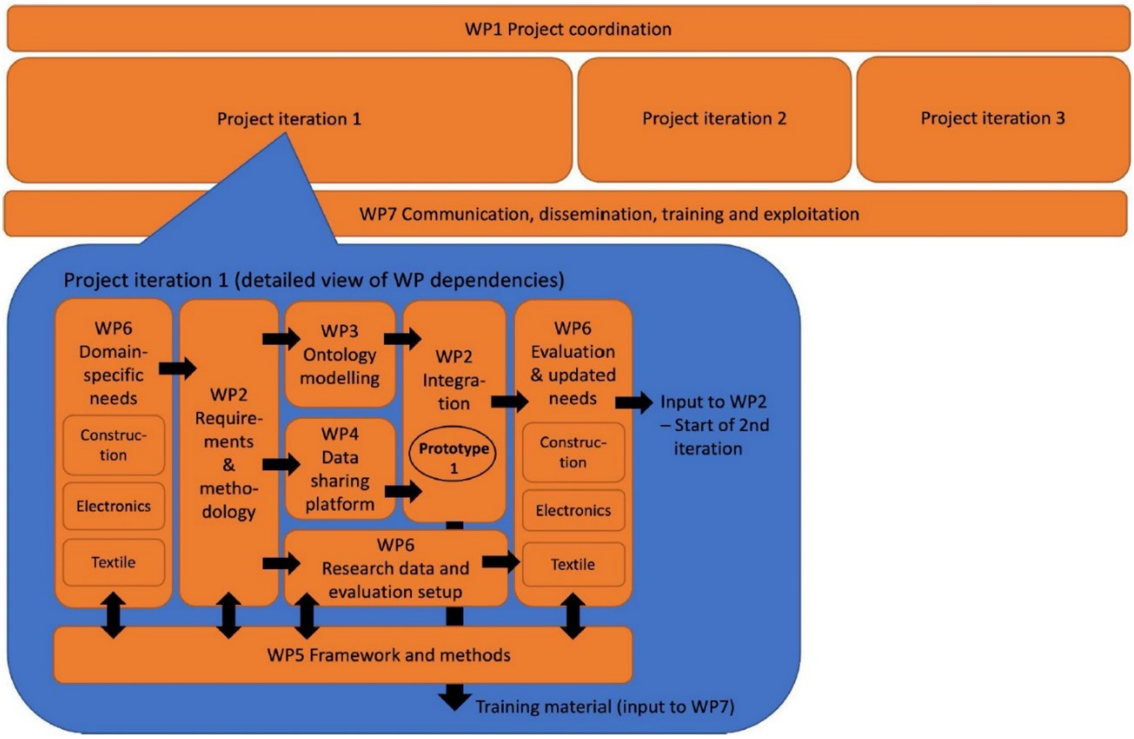
This project will develop at its basis a technology for allowing data sharing about materials and products at a global scale. Since access to verifiable information is central, the project will use well-established open standards for secure data and information sharing. As ownership and storage of data should remain with the actor that produces the data, a decentralised approach is necessary. Metadata and structures for transforming data into information (semantic descriptions and vocabularies) will be open, and comply with FAIR principles, to enable the highest possible degree of semantic interoperability and automation in data sharing. For sensitive data, methods allowing for proof of the existence of the data will be used, where these proofs can be shared while the actual data is kept private.

Further, this interdisciplinary project will also develop integrated tools and methods for further enhancing a Circular Economy. Although the importance of various 'flows' – resource flows (the various forms resources can take along their journey, e.g. material, component, product), information flows, energy flows, and value flows – has been widely acknowledged within the transformation to a Circular Economy, they have not been integrated or linked into a single framework or approach. Without such integration or linking it is not possible to make robust designs of circular value networks and to implement and operate value network coordination within industry.

The Onto-DESIDE project applies an iterative methodology, where research and innovation are driven by industry needs identified in a set of industry use cases, and solutions become more mature with each iteration. Three project use cases representing three distinct industry sectors – construction, electronics and appliances, and textile – will contribute to identify the needs and technical requirements of the Open Circularity Platform, but also act as test beds and evaluation scenarios for the novel solutions produced. This way, the project aims to show that the Open Circularity Platform is concrete enough to solve specific problems (i.e., the three specific use cases) but also has potential to be widely applied.

The project consists of three iterations, where each technical Work Package (WP) contributes to all the iterations. The WP dependencies are illustrated in Figure 1 where the details of the first project iteration are shown. The duration of the first project iteration is Month (M) 1-18, while the second and third iterations are shorter (M19-27 and M28-36 respectively). Each iteration ends with a collection of feedback from the industry use cases, analysed and reported in a WP6 evaluation report.

FIGURE 1- PROJECT OUTLINE AND DETAILED DEPENDENCIES BETWEEN WORK PACKAGES EXEMPLIFIED BY THE FIRST ITERATION



2.1 Objectives and Research Methodology

2.1.1 Objectives

Today, lack of support for sharing data in a secure, quality-assured, and automated way is one of the main obstacles that industry actors point to when attempting to create new circular value chains.

Work package 4 (WP4) will contribute to an open decentralised digital platform that enables secure collaboration. This includes supporting the correct enforcement of access control policies, as well as using verifiable credentials to prove the existence of sensitive data instead of publishing or sharing this sensitive data. The outcome of the WP will be an Open Circularity Platform, i.e., an open framework for secure and privacy-preserving digital and automated data sharing, which enables verifiable, traceable, and decentralised sharing of data expressed and documented using the ontologies from WP3, within and across industry domains.

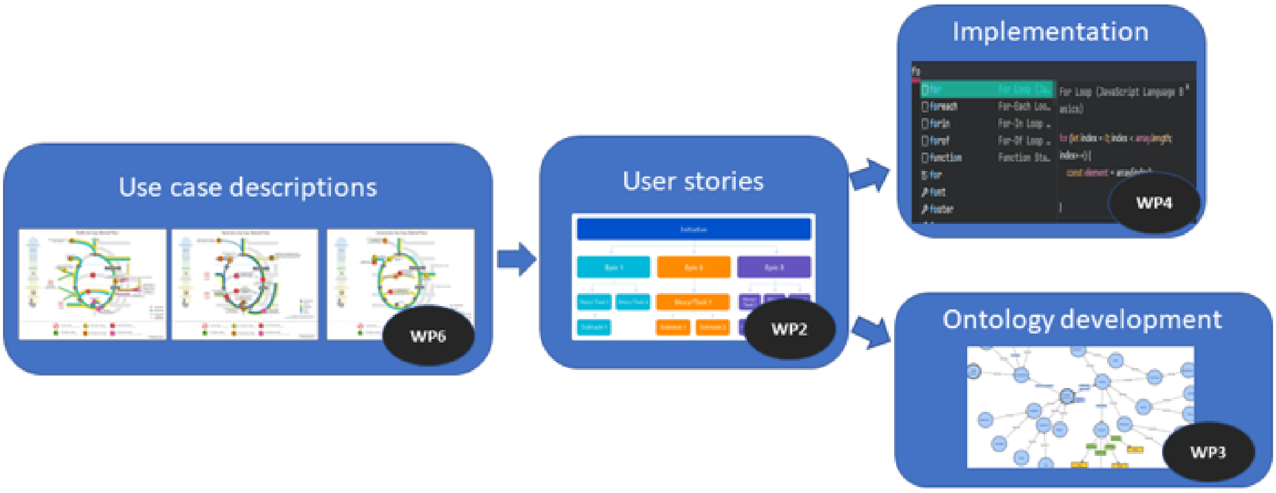
2.1.2 Research Methodology

The concrete research process of the Onto-DESIDE project has been divided into three iterations, each divided in 3 steps (cf. Figure 2):

- Step 1: analysis of needs and elicitation of requirements
- Step 2: research and technical development, including solution integration into a coherent prototype
- Step 3: use case-based observation and evaluation, providing feedback as well as revised and extended needs and requirements to start off the next iteration.

Specifically, for WP4, the focus is the technical development of the Open Circularity Platform, adhering to the technical requirements as put forward in WP2. For this, a new method will be devised to set-up decentralised networks of data vaults and actors.

FIGURE 2 DIAGRAM VISUALISING THE WORKFLOW OF WP4 AND THE RELATED ARTEFACTS



2.2 Tasks and Deliverables

WP4 is led by IMEC and is divided into five tasks, each related to the objectives of the work package. These tasks are outlined below:

- T4.1 - Data transformation - lead: IMEC, participants: CIRC
- T4.2 - Retrieving public and private data - lead: IMEC, participants: LIU
- T4.3 - Verifiable statements and credentials - lead: IMEC, participants: LIU, CIRC
- T4.4 - Blockchain-based implementation - lead: CIRC, participants: IMEC
- T4.5 - Querying data - lead: IMEC, participants: LIU, FAS

Two deliverables are to be produced for WP4 during the project:

- D4.1 Digital twin concept design, including ontology-based data sharing platform architecture and methodology (v1 M9, v2 M24, v3 M33) - report
- D4.2 Open circularity platform (v1 M10, v2 M22, v3 M31, v4 M33) - software

This document constitutes the report for deliverable D4.1 and aims to describe the Digital twin concept design, including ontology-based data sharing platform architecture and methodology. In this document, only T4.1, T4.2, and T4.5 are discussed: T4.3 and T4.4 are only recently started and will be part of v2 of the deliverable.

3 Digital Twin Concept Design

While semantic interoperability and ontology-based data documentation are essential enablers for large scale digital Circular Economy, it is not enough in itself. Semantically described data also needs to be put into use, in automated processes. Today, there is limited data collaboration within industry domains and even less across domains, and as a result, new circular value networks are only created between known actors that have a certain degree of comfort working together[1] - limiting the possibilities of more and more high value circulation scenarios. Open collaboration could remedy this, but data and ontologies cannot solve the problem alone. To facilitate open collaboration in a data driven circular economy a new entity is needed; the digital twin of circularity.

Here we rely on the digital twin definition provided by the Digital Twin Consortium[2], part of the international standards organisation, the Object Management Group[3]. The definition states that a digital twin

- is a synchronised virtual representation of real-world entities and processes;
- uses real-time and historical data to represent the past and present and simulate predicted futures;
- transforms businesses by accelerating holistic understanding, decision-making, and effective action; and
- is motivated by outcomes, tailored to use cases, powered by integration, built on data and guided by domain knowledge.

Digital twins built upon a shared vocabulary – i.e. defined in an ontology network – are reusable as templates for a certain type of circular value network, and could at minimal effort be shared with a different set of actors or used within a different industry domain to instantiate new value networks.

However, instead of having each industry domain create their own circular interconnections over time, the core logic of circularity should be common and manifested in a digital entity that translates between industry domains. By translating between domains, there is no need for a central information repository: every organisation will keep and manage their own data. By building on well-established standards for semantically describing, interlinking, and sharing data, collaboration is made secure and scalable. On its own, the twin represents value to stakeholders by providing the technical infrastructure for making data exchange in complex circular eco-systems manageable and reusable.

Technically, we will implement the digital twin idea as an open circularity platform using existing and emerging Web technologies, such as RML for semantically annotating heterogeneous data sources[4], Solid for building decentralised applications based on Linked Data principles[5], and incorporating validation and verification methods that provide proofs of data authenticity[6]. Given a commonly

[1] Bressanelli, G.; Adrodegari, F.; Perona, M.; Saccani, N. Exploring How Usage-Focused Business Models Enable Circular Economy through Digital Technologies. Sustainability 2018, 10, 639.

[2] https://www.digitaltwinconsortium.org/initiatives/the-definition-of-a-digital-twin.htm

[3] https://www.omg.org/

[4] Dimou, A.; Vander Sande, M.; Colpaert, P.; Verborgh, R.; Mannens, E. & Van de Walle, R. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. Proceedings of the 7th Workshop on Linked Data on the Web, CEUR- WS.org, 2014, 1184

[5] Solid Community Group. Solid Technical Reports. World Wide Web Consortium (W3C), 2021

[6] Burnett, D.; Noble, G.; Zundel, B.; Longley, D. & Sporny, M. Verifiable Credentials Data Model 1.0. World Wide Web Consortium (W3C), 2019

understood ontology, we foresee the following three challenges to provide the novel decentralised solution:

- semantic interpretation of existing data, so that actors can rely on existing infrastructure;
- a decentralised network to publish and retrieve semantically annotated data, behind a layer of authentication and authorization, so that actors can share their data with only those partners they are comfortable with; and
- a verification method so that collaborating actors can trust the data they are using.

Our objective is to demonstrate (at TRL6) an ontology-based decentralised data sharing platform that operationalises the digital twin idea: maximally taking advantage of existing IT infrastructures and standards, without compromising on access control and trust.

4 Methodology

We will set up an ontology-based decentralised data sharing platform using the Solid protocols. As no best-practice methodology for designing such a data sharing platform is in place, part of our research is to devise such a methodology.

4.1 Proposed Methodology

We present following methodology:

1. Extract use cases by means of user stories. A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer. *This allows us to form the (high-level) requirements.*
2. Create a list of data sources per use case: what is their content, how do you access it, and who owns them. Use cases can involve different types of data, and different ways of governance. As such, multiple data sources can provide essential or context information concerning a single use case. Whether – and how – this data must be interpreted semantically, depends on the use case requirements. *This allows us to identify what existing or sample data is available.*
3. Prioritise the use case based on data availability, stakeholder importance, and complexity. *Based on this prioritisation, we can select the first use case to work on and continue iteratively.*
4. Create a list of technical scenarios within the use case. A technical scenario is a list of steps, written in plain text, where detailed data flows are elaborated on specific use cases. E.g., when a use case specifies "I want to retrieve product information from the manufacturer.", a technical scenario specifies which specific product information is retrieved from the manufacturer, in which required format. *This allows us to further detail the technical data requirements.*
5. Per scenario, create a list of actors and existing systems (databases, APIs, user interfaces, and so on), and how the data flows between them. This could be via a data flow diagram, or a (more detailed) sequence diagram. These diagrams provide details about the calls that happen between the different actors and components.
6. Devise (alternative) Solid-based set-ups that adhere to the requirements and data flows.

4.2 Discussion

Within Onto-DESIDE, the first step is achieved via a combination of defining example cases (WP6) and technical requirements (WP2). WP6 first delivered Deliverable 6.1, a set of example cases spanning different industries (electronics & appliances, textiles, and construction) allowing for the testing and validation of the solutions developed within the Onto-DESIDE project. Based on Deliverable 6.1, WP2 condenses the use cases into technical requirements serving as the main input for this deliverable, allowing us to identify the technical components per user story. An overview can be found in Appendix A.1 Table: User story alignment.

In anticipation of Deliverable 2.1 (WP2), we considered the example cases in Deliverable 6.1 to kickstart the initial research and development iteration within WP4. Thereupon, when the user stories in Deliverable 2.1 are updated, we prioritise the user stories based on the feasibility of implementation and blocking factors. Subsequently, we can decide to follow a breadthwise (considering more user stories) or depth-wise approach (considering more details of the selected user stories).

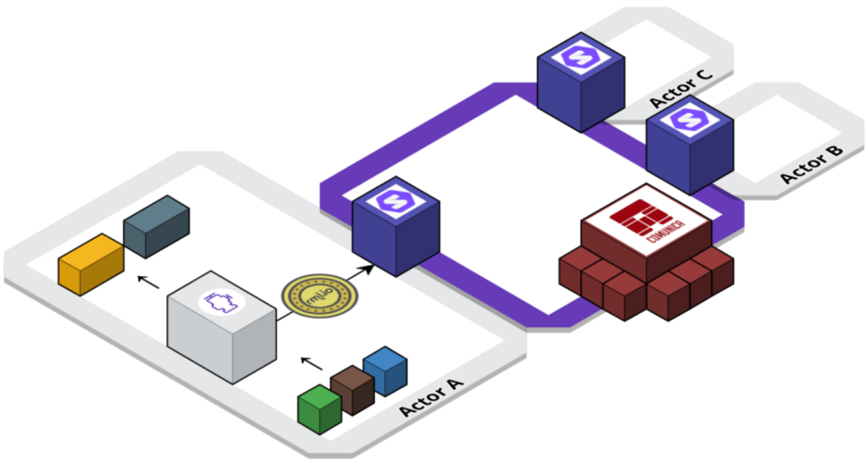
The prioritisation of the user stories is based upon the implementation feasibility at the time of the development iteration, which depends on data availability, existing technologies, tooling, and in-house knowledge. Simultaneously, this prioritisation allows us to identify knowledge gaps and determine research directions.

The Construction use case as specified in Deliverable 6.1 serves as a starting point for devising a minimal architecture covering the main technical components required to obtain a decentralised data sharing platform that allows for the integration of digital twins, and iteratively evaluate and extend the architecture as needed to further align with the user stories.

5 Ontology-based Data Sharing Platform Architecture

In the following we elaborate on i) how to incorporate a Linked Data based Digital Twin into a decentralised data sharing platform, ii) how the Solid ecosystem is leveraged to establish a decentralised data sharing platform that connects these Digital Twins to enable open collaboration, and (iii) how to uniformly access that decentralised data using federated querying.

Figure 3 Diagram visualising a complete ontology-based data sharing platform



5.1 Linked Data-based Digital Twins (T4.1)

Within an individual company, existing data needs to be semantically annotated to conform to the concepts of the ontology network. The connection from existing companies to an ontology-based data sharing platform requires mapping their existing data to the ontologies used in the network. Requiring each partner to design their own custom annotation application would place a heavy burden and inhibit uptake. Instead, we leverage mapping languages to declaratively describe how the source data maps to semantically enriched RDF data that conforms to the concepts of the ontology network developed by WP3.

A mapping language describes different types of mappings between a source schema and a target schema and provides a means for linking a particular data source to its specific mapping policy. Various mapping languages, exhibiting different characteristics and features, exist[7]. We leverage the RDF Mapping Language (RML), an extension of W3C's recommended R2RML[8], supporting heterogeneous data sources, hence, allowing actors to maximally rely on existing systems. Multiple RML processors can be used for generating RDF. We selected the RML Mapper[9], a fully spec-compliant processor that can be used in any development environment installed with Java.

[7] Van Assche, D.; Delva, T.; Haesendonck, G.; Heyvaert, P.; De Meester, B. & Dimou, A. Declarative RDF graph generation from heterogeneous (semi-)structured data: A systematic literature review. Journal of Web Semantics, 2022

[8] S. Das, S. Sundara, and R. Cyganiak, “R2RML: RDB to RDF Mapping Language,” World Wide Web Consortium (W3C), Working Group Recommendation, Sep. 2012. [Online]. Available: http://www.w3.org/TR/r2rml/

[9] https://github.com/RMLio/rmlmapper-java

However, instead of directly describing the transformations in RML we leverage YARRRML, a user-friendly YAML syntax for describing RML mappings, hence lowering the technical barrier for actors[10]. Typically, RML processors do not natively support the execution of YARRRML mappings, therefore, these YARRRML mappings need to be converted to RML first, for which we use the YARRRML parser[11].

5.2 Decentralised Data Sharing Platform (T4.2)

Across companies, a network of decentralised data stores that allow fine-grained access control is needed. Traditional, centralised platforms require centralised systems for identification, authentication, authorization, and data storage. This requires actors to use the systems that have been put forward by the platform itself. For example, actors are required to store their data on the centralised platform servers, thereby giving up control of their data to the platform. In contrast, decentralising every part of a platform enables actors to choose which solutions to use for identification, authentication, authorization, and data storage.
Solid is a novel concept that aims to radically change the way Web applications work today. The Solid ecosystem encapsulates a set of W3C standards and tools, based upon the Linked Data principles, taking authentication and authorization into account, and aiming towards a sustainable Web and decentralised data-ecosystem[12]. By drastically separating data from logic, services and applications become federated views on top of a set of distributed data pods, and service providers no longer need to centralise all data themselves.

In Solid, decentralisation does not only pertain to where data is stored, but also to every other component within the Solid ecosystem. As such, Solid constitutes of separate interoperable standards covering identification, authentication, authorization and managing resources.

5.2.1 Building Blocks

Data collaboration is vital to a digital Circular Economy as it enables actors to securely publish, share, and retrieve the semantically annotated data within the circular network of digital twins. In the following we will discuss the essential components required for actors to securely and confidentially operate on the decentralised data.

Identity. Users, organisations, services, and applications of the decentralised data-sharing platform need to be unambiguously identifiable. Such *identity resources* are the cornerstone to enabling trust between actors (authentication), defining access control (authorization), and determining each actor's associated decentralised data store(s) to enable data collaboration. The WebID specification allows – through a Web resource – to describe the identities of users, organisations, services, and applications[13].

Authentication. To establish trust between actors, authentication is required to verify the identity of each party. The current Solid ecosystem recommendation for authentication relies on Solid-OIDC:

[10] Heyvaert, P.; De Meester, B.; Dimou, A. & Verborgh, R. Declarative Rules for Linked Data Generation at your Fingertips! The Semantic Web: ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers, Springer, Cham, 2018, 11155
[11] https://github.com/RMLio/yarrrml-parser
[12] Verborgh, R. Paradigm shifts for the decentralized Web 2017
[13] A. Sambra, H. Story, and T. Berners-Lee, "WebID 1.0," W3C WebID Community Group, W3C Editor's Draft, Mar. 2014. [Online]. Available: https://www.w3.org/2005/Incubator/webid/spec/identity/

an OpenID-Connect extension to authenticate users without hard-coding the connection between the Identity Provider and the Data Store[14].

Authorization. Access Control defines which data can be accessed by what or whom. The Access Control Policy (ACP) specification describes how authorised access can be defined by associating access permissions with identities[15].

Resource management. When separating data from applications, strict protocols need to be in place to manage resources. Basic data manipulation is described by the Linked Data Platform (LDP) specification, where actors can manage and operate on both binary data (e.g., PDF files) and semantically annotated linked data (e.g., RDF graphs).

Data services. Interacting with the data in a decentralised network requires data services. This could be the end-user applications that directly operate on the data, or middleware-like services. Complex data retrieval operations can be expressed using the SPARQL protocol[16].

Multiple Solid server implementations are available. We make use of the Community Solid Server[17]: the reference open-source implementation.

5.2.2 Solid Servers and Pods

Users (within the context of this project also referred to as actors) can store their data in one or more data stores (called pods) that are fully controlled by their respective user.

Solid servers host one or more Solid pods in which actors can store any kind of data. Solid servers can be hosted by a third party (i.e., a Pod provider), or can be self-hosted. The former option allows the actor to delegate some technical workload to the Pod provider, while the latter provides the actor with the highest level of control. Therefore, we designed the infrastructure of the data-sharing platform so that each Solid pod is hosted by, and thus under complete control of, the actor itself. As such, actors can maximally rely on their existing infrastructure and are alleviated from potential conflicts with Data Processing Agreements that may elicit from storing data on third party infrastructures. Moreover, the current set-up does not preclude the usage of a Pod provider: these can be introduced in the network in a later stage.

[14] A. Coburn, e. Pavlik, and D. Zagidulin, “Solid-OIDC,” Solid Community Group, Editor’s Draft, Mar. 2022. [Online]. Available: https://solidproject.org/TR/oidc

[15] “Access Control Policy (ACP),” Solid Community Group, Sep. 2022. [Online]. Available: https://solid.github.io/authorization-panel/acp-specification/

[16] C. B. Aranda et al., “SPARQL 1.1 Overview,” World Wide Web Consortium (W3C), Recommendation, Mar. 2013. [Online]. Available: http://www.w3.org/TR/sparql11-overview/

[17] https://github.com/CommunitySolidServer/CommunitySolidServer

5.3 Decentralised Querying (T4.5)

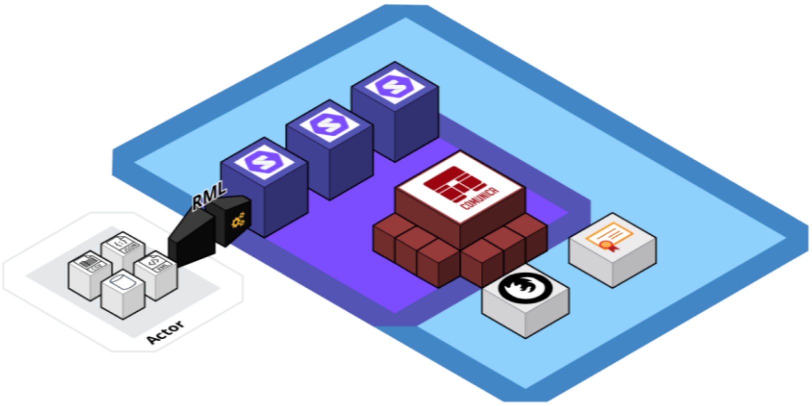
From an application point of view, uniform access to this decentralised data is needed[18]. We employ the state-of-the-art federation engine Comunica[19] to transparently query decentralised data pods behind an authentication layer: Comunica has native Solid authentication support[20].

Using federated querying allows us to demonstrate how an application would interact with the decentralised network using a uniform access method. When the functionalities of the network will change (e.g., support for verifiable credentials by M24), this uniform access method will not need to change.

6 Proof-of-Concept Implementation

In this section, we detail the implementation of version 1 of the proof-of-concept. We detail the example use case, after which we align it with the devised methodology and concrete extracted use cases. More details on the technical implementation are provided in Appendix A.2 Technical overview, a complete technical description will be made available as part of Deliverable 4.2.

FIGURE 4 DIAGRAM VISUALISING THE PROOF-OF-CONCEPT IMPLEMENTATION



6.1 Example Use Case: Construction

We use an example use case to identify which technical components are required and how these components interact.

We select a use case covering a generalised set of actors (i.e., manufacturer and user) and concepts (i.e., product, material, and location).

The goals for this iteration are the following:

[18] R. Verborgh, "Reflections of knowledge." Dec. 2021. [Online]. Available: https://ruben.verborgh.org/blog/2021/12/23/reflections-of-knowledge/

[19] https://comunica.dev/

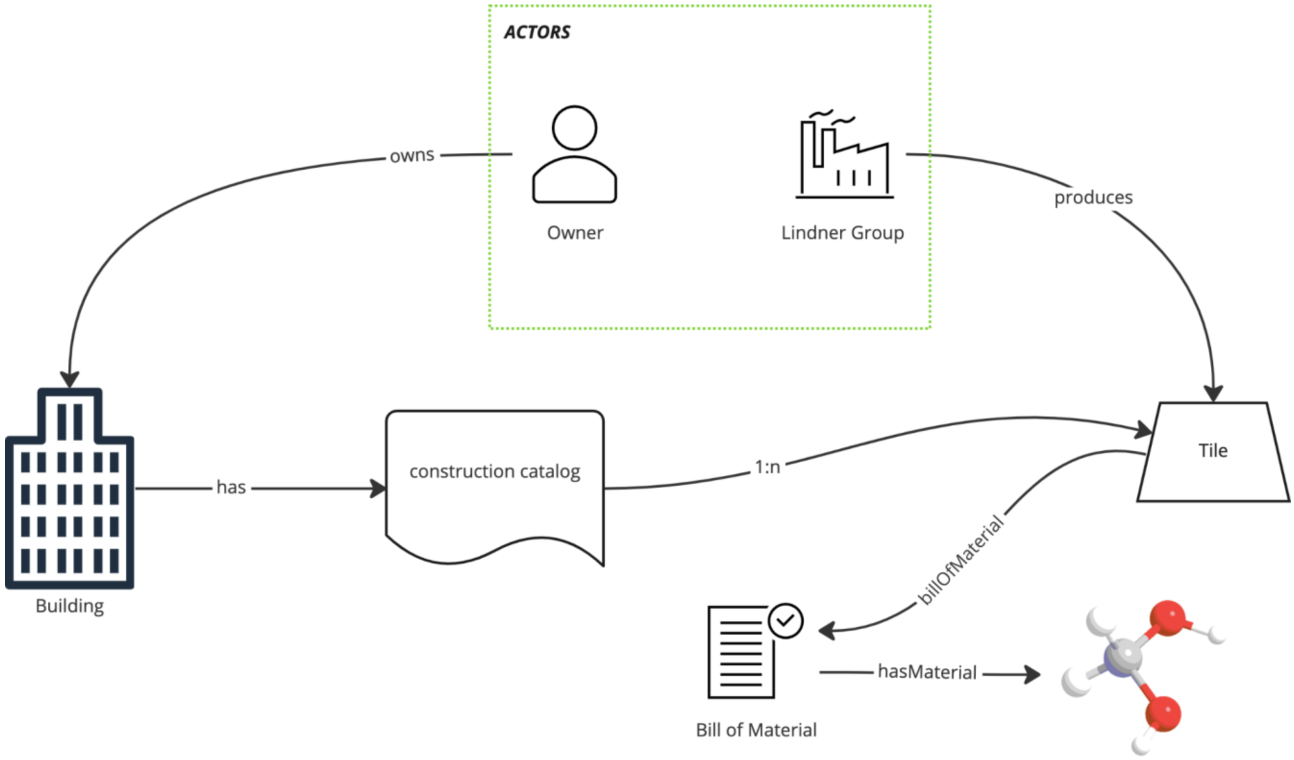
[20] https://comunica.dev/docs/query/advanced/solid/

- Each actor can be uniquely identified and has a data pod connected to the decentralised data sharing platform.
- Each actor's data can be transformed to RDF and stored to the actor's data pod.
- The data of each actor can be queried.

The use case is as follows:

- A building owner owns some buildings.
- Floors in a building are covered with floor tiles, delivered by a manufacturer, who uses materials to construct the tiles.
- At some time in a building's lifetime, the building owner refurbishes a building and wants to decide on what to do with the floor tiles, based on economic and environmental costs.

FIGURE 5 DIAGRAM VISUALISING THE EXEMPLARY USE CASE



At the time of writing, there is no actual or sample data available for any of the user stories. Therefore, we create minimal dummy data for each actor that is sufficient to demonstrate the technical implementations.

For demonstration purposes, we created example data that serves as the source data originating from two actors present in the example use case: the *building owner* and a manufacturer, the *Lindner Group*.

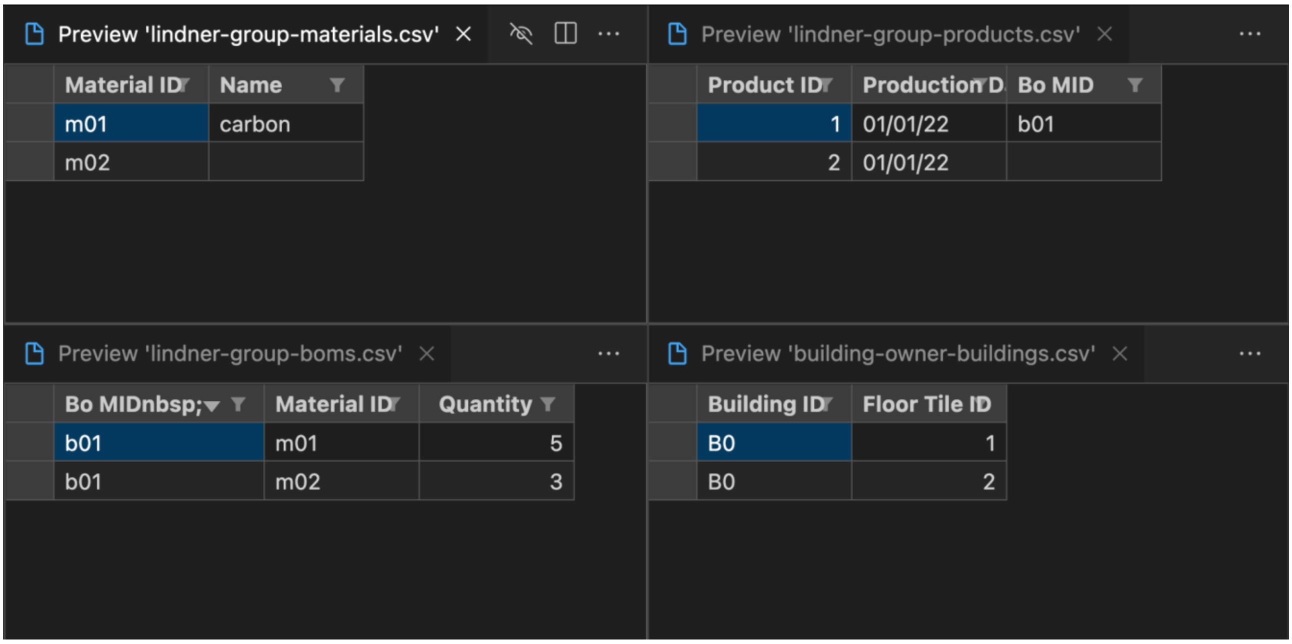
The figure below illustrates the example data:

- For the building owner:
  - An inventory of buildings, containing per building the used tiles: "building-owner-buildings.csv".
- For Linder Group
  - An inventory of products (tiles), containing per product the production dates and the bill of materials: "lindner-group-products.csv".

- An inventory of bills of materials, containing per material the quantity used: “lindner-group-boms.csv”.
- An inventory of materials, containing per material its name: “lindner-group-materials.csv”.

It is assumed that the tile IDs in the building owner's CSV are aligned with the product IDs of the floor tile producer.

FIGURE 6 PREVIEW OF THE DIFFERENT DECENTRALISED DATASETS


Preview 'lindner-group-materials.csv'

Material ID	Name
m01	carbon
m02	

Preview 'lindner-group-products.csv'

Product ID	Production D	Bo MID
1	01/01/22	b01
2	01/01/22	

Preview 'lindner-group-boms.csv'

Bo MIDnbsp;	Material ID	Quantity
b01	m01	5
b01	m02	3

Preview 'building-owner-buildings.csv'

Building ID	Floor Tile ID
B0	1
B0	2

We validate the initial development iteration by verifying that we can query each actor’s data. More specifically, our queries are detailed below.

- What are the *tiles* of building B0?
- What are the *materials* of the product with Product ID 1?

To realise this, we make use of the following components.

We map the raw input data (.csv files) into linked data using the YARRRML parser and the RML mapper. The result is a set of Turtle files: “buildings.ttl” for the building owner; “products.ttl”, “materials.ttl” and “boms.ttl” for the Lindner Group.

Both actors have their own pod in their own Solid server, here a Community Solid Server. Each pod contains the Turtle files of the actor owning the pod. To interact with the Solid Server, e.g., putting the data as turtle files in place, we make use of Bashlib: a Command-Line Interface to interact with actors in a Solid network[21].

To query the distributed data, we use Comunica: a tool that can issue SPARQL queries over multiple data sources (federated queries). We can demonstrate this visually using this web client for Comunica[22].

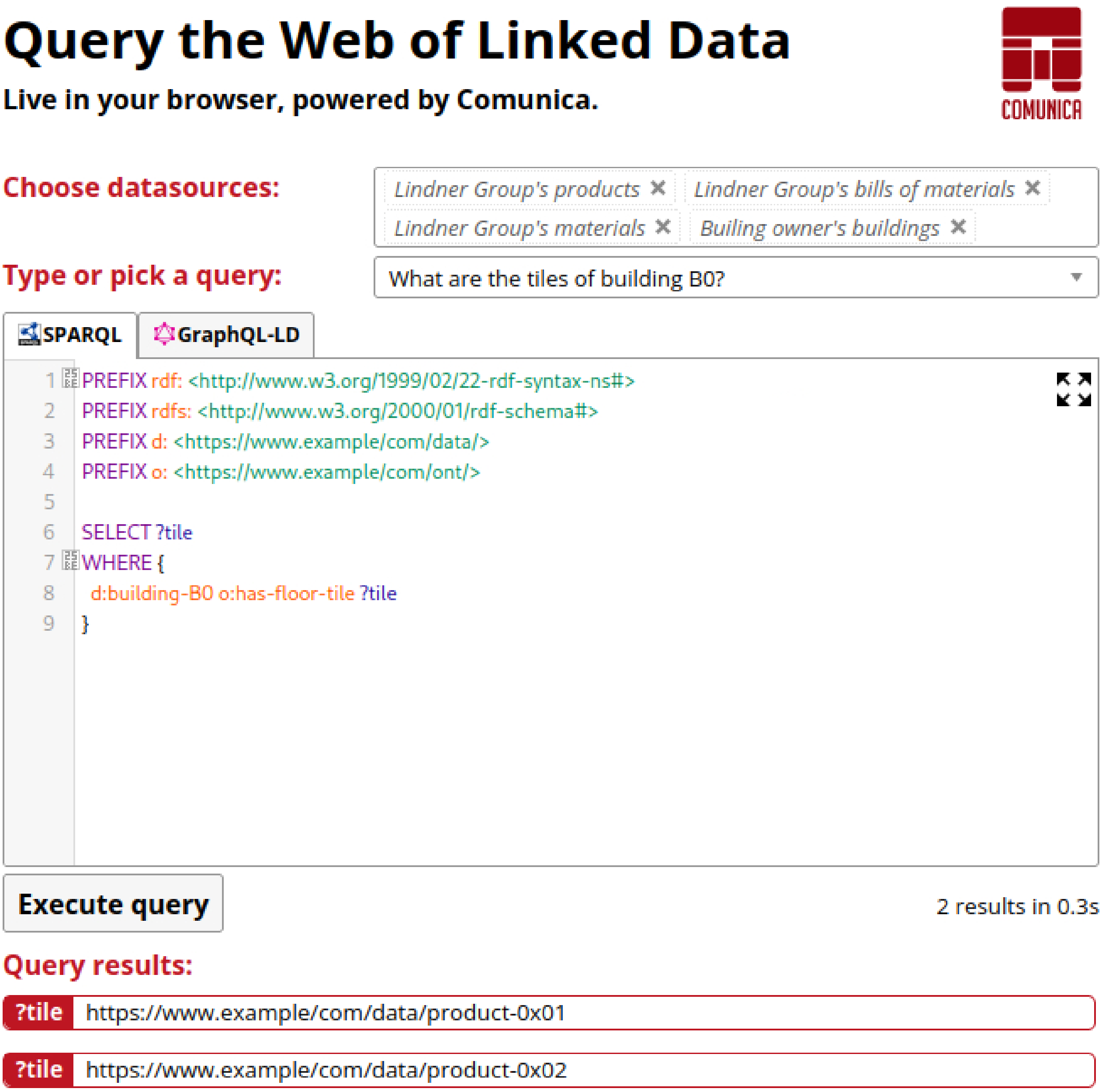
[21] https://github.com/SolidLabResearch/Bashlib
[22] https://github.com/comunica/jQuery-Widget.js

We obtain the following results:

The results of the SPARQL query for “What are the *tiles* of building B0?” are shown in Figure 7.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX d: <https://www.example/com/data/>
PREFIX o: <https://www.example/com/ont/>

SELECT ?tile
WHERE {
  d:building-B0 o:has-floor-tile ?tile
}
```

FIGURE 7 QUERY RESULTS FOR THE QUERY "WHAT ARE THE TILES OF BUILDING B0?"

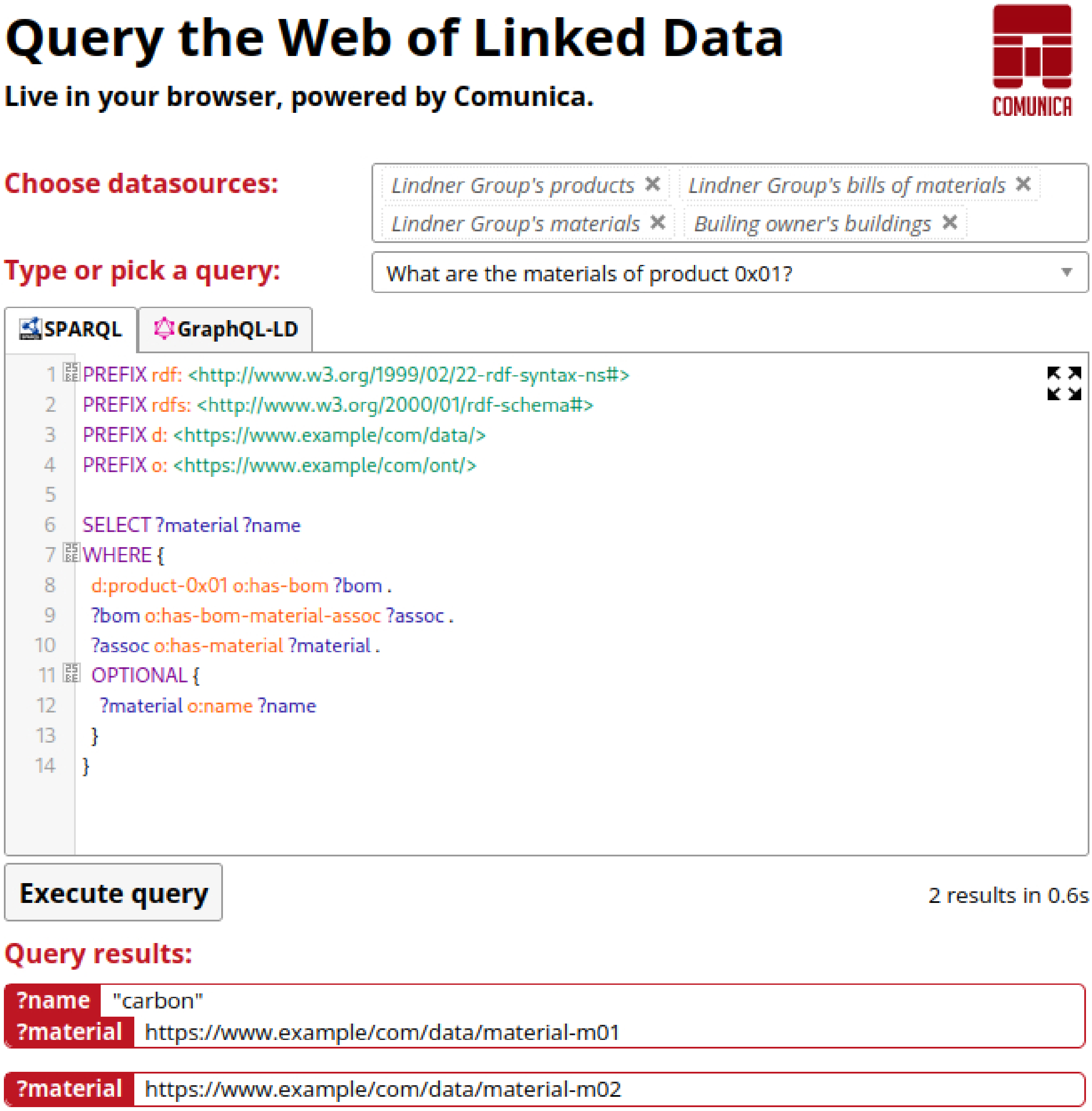


The results of the SPARQL query for "What are the materials of product 0x01?" are shown in Figure 8.

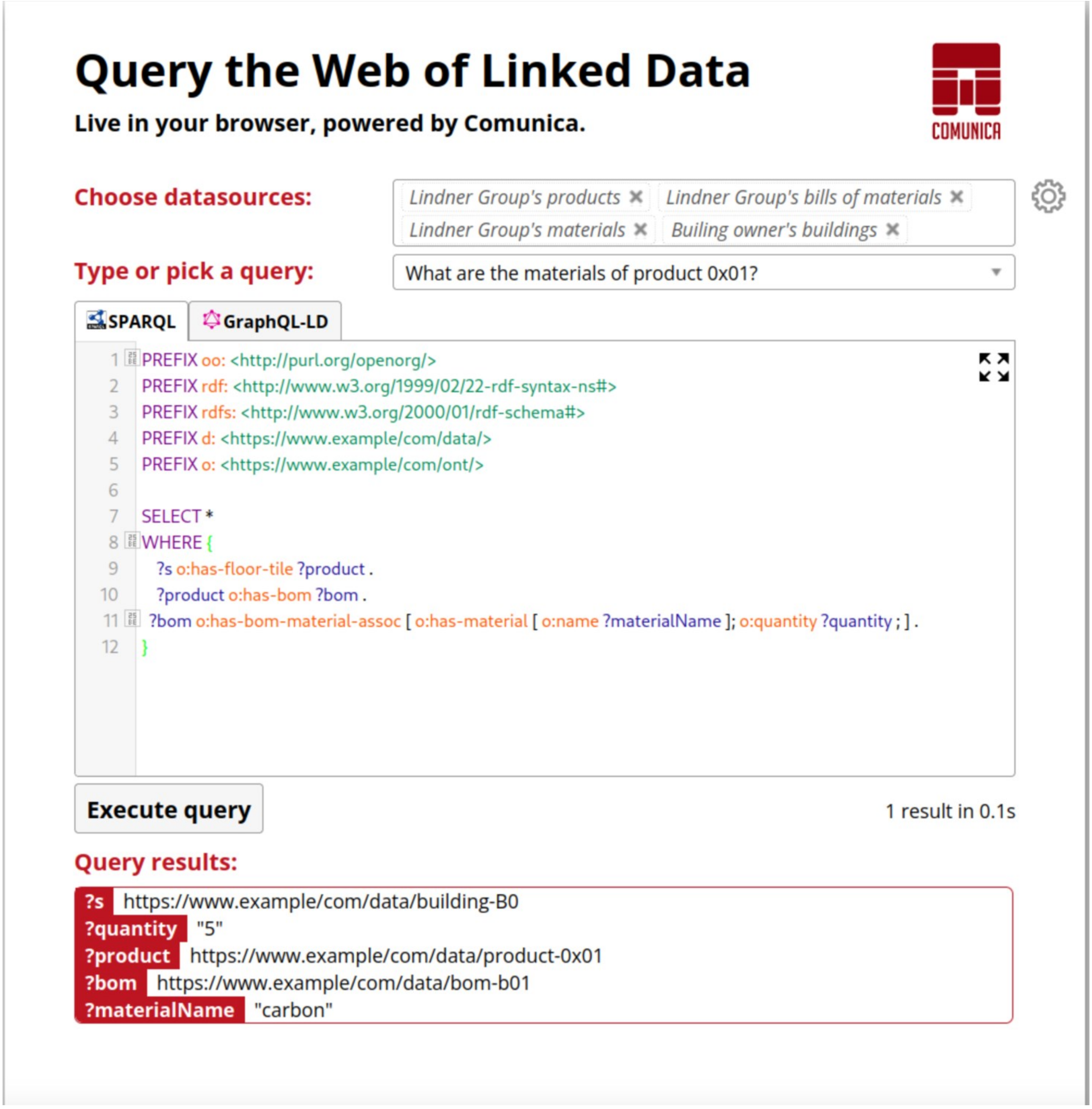
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX d: <https://www.example/com/data/>
PREFIX o: <https://www.example/com/ont/>

SELECT ?material ?name
WHERE {
  d:product-0x01 o:has-bom ?bom .
  ?bom o:has-bom-material-assoc ?assoc .
  ?assoc o:has-material ?material .
```

```
        OPTIONAL {
            ?material o:name ?name
        }
    }
```

FIGURE 8 QUERY RESULTS FOR THE QUERY "WHAT ARE THE MATERIALS OF PRODUCT 0X01?"



As a third example, we execute a manually defined query (hence, not a predefined query) that links building tiles to its composite materials through the Bill of Materials. This third example actively requires the integration of data from the building owner with data from the floor tile producer. As such, this query shows how this system is capable of answering questions over decentralised data stores.



6.2 Alignment with Concrete User Stories

The above-mentioned proof-of-concept implementation aligns with demonstrating the methodology as presented in Section 4.1:

1. In Deliverable 6.1, use cases are noted, and further detailed into concrete user stories. The generality of the simplified example use case allows for alignment with concrete use cases involving i) a user and manufacturer as actors; ii) product, material, and location as concepts; in which actors want to create a semantically annotated representation of their source data, which is uniformly accessible to actors within the decentralised data-sharing platform. For example, within the construction industry, we can align with CUS3 and CUS8.
2. Given there was no real-world data available yet, step 2 of the proposed methodology (detailed in Section 4.1, "Create a list of data sources per use case") was skipped.

3. We prioritised the use cases on increasing complexity, i.e., the more they aligned with the example use case as detailed in Section 6.1, the lower their complexity was estimated.
4. We further detailed the scenarios within the use case, and translated them into technical requirements, e.g., being able to ask the query “What are the tiles of building X?”
5. Based on the use case as visualised in Figure 5, we created a list of actors, existing systems, and how the data flows between them.
6. The proof-of-concept implementation can provide the answers to the questions as devised in step 4, using the actors and data flows as devised in step 5.

7 Conclusion

This deliverable contains four main parts covering the Digital twin concept design, the Solid set-up methodology, the ontology-based data sharing platform architecture, and proof-of-concept implementation.

Based on mocked data, we were able to validate two user stories (CUS3 and CUS8) as presented in Deliverable 2.1 using the components that we aim to use in the final demonstrator. Although this first proof-of-concept is very basic (no authentication, no support for verifying the answers), the functionality of working on top of existing (non-RDF) data in a decentralised setting using Solid pods is already showcased.

In the next step, we will complete the technical documentation of version 1 of the demonstrator (Deliverable 4.2), after which we will further align with the different (prioritised) use cases, provide more granular authorization support, and include verifiable credential support.

An important next focus point is the alignment with real-world datasets. As product data is typically very heterogeneous, not always well-structured, and sometimes contains intellectual property of the manufacturer, it became clear that raw product data is not easy to gather. A mitigation is to rely on data as specified in the existing regulatory framework, i.e., the type of data that is required during certification processes. In the next iteration of our demonstrator, we aim to align with real-world data by means of the certification data requirements, starting with the minimal value data sample, and aligned with the ontology network as designed in Work Package 3.

Our current solution materialises each actor’s source data to RDF and is subsequently stored on their Solid pods. Further research is needed towards creating a RDF representation in a virtualized manner.

8 Appendix A

8.1 A.1 Table: User story alignmentUser story alignment table

ID	Related technologies
CUS1	data querying (SPARQL + Comunica)
CUS2	data querying (SPARQL + Comunica)
CUS3	access control (Solid) data querying (SPARQL + Comunica)
CUS4	access control (Solid) data querying (SPARQL + Comunica)

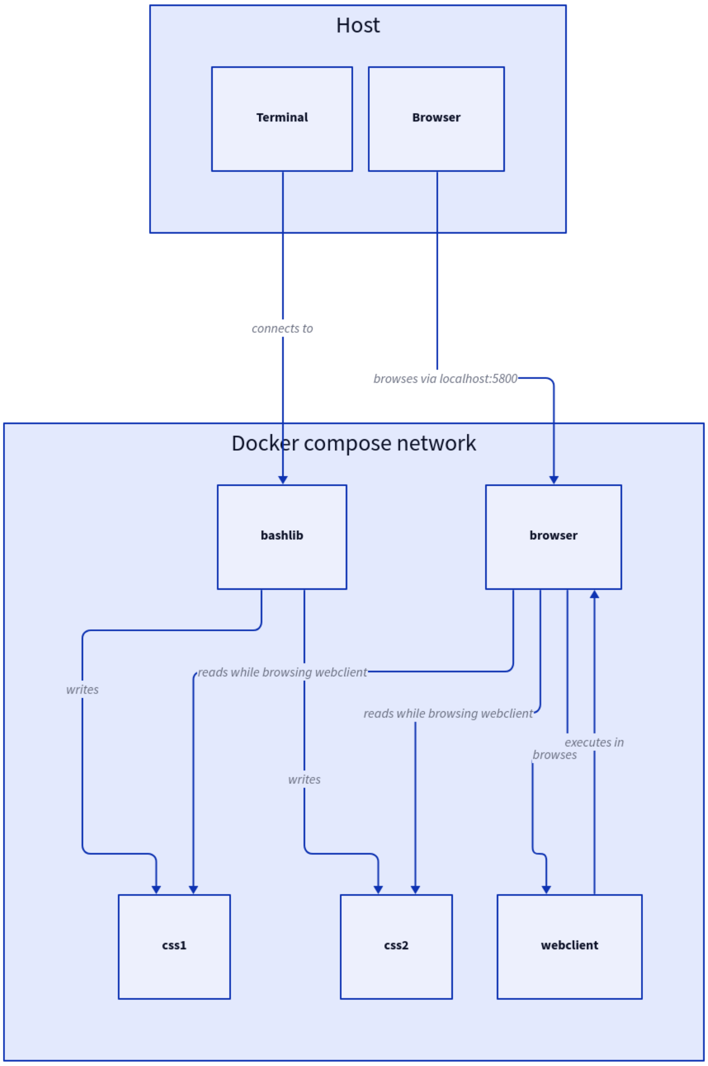
CUS5	access control (Solid) data querying (SPARQL + Comunica)
CUS6	access control (Solid) data querying (SPARQL + Comunica)
CUS7	access control (Solid) data querying (SPARQL + Comunica)
CUS8	access control (Solid) data querying (SPARQL + Comunica)
CUS9	access control (Solid) data querying (SPARQL + Comunica)
CUS10	access control (Solid) data querying (SPARQL + Comunica)
CUS11	access control (Solid) data querying (SPARQL + Comunica)
CUS12	access control (Solid) data querying (SPARQL + Comunica)
CUS13	access control (Solid) data querying (SPARQL + Comunica)
EUS1	provenance trails, VC
EUS2	provenance trails
EUS3	certification, VC
EUS4	access control (Solid) data querying (SPARQL + Comunica)
EUS5	access control (Solid) data querying (SPARQL + Comunica)
EUS6	access control (Solid) data querying (SPARQL + Comunica)
TUS1	UI + Forms. Data querying (SPARQL + Comunica)
TUS2	
TUS3	UI, Linked Data Notifcations (Solid)
TUS4	material descriptions, process descriptions, trustful data access
TUS5	data transformation (RML)
TUS6	access control, trust, VC (Solid) data querying (SPARQL + Comunica)
TUS7	access control (Solid) data querying (SPARQL + Comunica)
TUS8	access control (Solid) data querying (SPARQL + Comunica)
TUS9	Certification, VC
TUS10	access control, trust, VC (Solid) data querying (SPARQL + Comunica) Rules to define minimization strategies?
TUS11	authentication, access control, trust, VC (Solid)
TUS12	ui
TUS13	ui, querying, RML, automated lifting process (e.g. on source data change)
TUS14	ui, querying, process descriptions, provenance trails (for remanufactured products)

TUS15	access control (Solid) data querying (SPARQL + Comunica)
TUS16	access control (Solid) data querying (SPARQL + Comunica)
TUS17	Certification, VC, Trust
TUS18	access control (Solid) data querying (SPARQL + Comunica)
TUS19	access control (Solid) data querying (SPARQL + Comunica)
TUS20	access control (Solid) data querying (SPARQL + Comunica)
TUS21	access control (Solid) data querying (SPARQL + Comunica)
TUS22	access control (Solid) data querying (SPARQL + Comunica)
TUS23	access control (Solid) data querying (SPARQL + Comunica)

8.3 A.2 Technical overview

This appendix provides a technical overview of how the different components are integrated in an end-to-end demonstrator, with a focus on reproducibility, i.e., making it easy to reproduce this network locally. Below, we first detail the different involved high-level components, and second the use case flow. Please note that this only gives a short overview. More details and reproducible code on the technical setup will be provided as part of Deliverable 4.2.

FIGURE 9 TECHNICAL OVERVIEW OF THE DIFFERENT COMPONENTS AS SET UP IN THE PROOF-OF-CONCEPT IMPLEMENTATION



We use Docker Compose[23] to configure the multi-container architecture that represents the Solid-based decentralized data sharing platform. The configuration is as follows:

- Service `css1` runs Community Solid Server (CSS), hosts the pod of Lindner Group.
- Service `css2` runs another CSS, hosting the pod of the building owner.
- Service `bashlib` runs a Linux container in which Bashlib is used to operate on Solid pods (e.g. setup, populating the Solid pod with the generated RDF data).
- Service `webclient` runs the Comunica Webclient, providing a Web UI to execute queries on Solid pods.

[23] https://docs.docker.com/compose/

- Service `browser`, a Firefox container providing a means to browse the Solid-based data-sharing platform.

During the setup-flow, an administrative user operates the bashlib service to load the generated RDF data into a Solid pod. For the sake of simplicity, a single bashlib service is employed to operate both actor's Solid pods. Note, however, that this can easily be extended to using one bashlib service per actor.

During the usage-flow, an end user browses to the emulated Firefox browser (http://localhost:5800/) which provides access to the Solid-based decentralised data-sharing platform. Within the emulated browser, the user navigates to the Comunica Webclient (http://webclient) which provides a set of predefined queries the user can execute over the Solid pods.