Deliverable 2.2

Design Specification

Verein zur Förderung der selbstständigen Nutzung von Daten
2540 Bad Vöslau
ZVR: 789007092

Contact: office@ownyourdata.eu

Content

1 Introduction

The deliverable describes the design specification of the FFG funded IDunion project. The document introduces first the individual building blocks and describes those in detail with a specific focus on new functionality created in the course of the project. Chapter 3 outlines the interfaces and provides a detailed outline of the workflow for the three main use cases demonstrated in the project:

- creating a verifiable credential
- verifying a credential
- sharing personal data with other institutions

Based on components in chapter 2 and dataflows in chapter 3 the solution is verified through test procedures as described in chapter 4 together with the test system setup.

The document concludes with an outlook about further development beyond the funding provided by FFG.

1.1 Background

Currently, vaccination and immunisation information are spread over different organisations like labs and hospitals as well as pharmaceutical companies together with government agencies. A patient previously only had a paper certificate that provides vaccination treatments with often difficult to read handwritten additional information. Through the corona pandemic digital immunisation passports became more common and several concerns and challenges arised.

The main focus of this project is to address questions on Data Interoperability & Compatibility through establishing interfaces between the health industry and individuals as well as pushing forward on standardised interfaces for Personal Data Stores. Additionally, we address Data Transparency (Usage Policies and Data Provenance in Semantic Containers) and Security & Privacy (by applying blockchain technology and digital watermarking on data sharing).

1.2 Relation to other deliverables

This design document is one out of two documents providing the detailed description about this project:

- D2.1 Requirements Document: lists functional and non-functional requirements for creating and verifying credentials, as well as sharing data between individuals and organisations
- D2.2 Design Specification: describes and depicts the system design together with API endpoints and data formats of the various components

2 Functional components

This section describes new developments in the course of the project for the functional components of the IDunion project as depicted in Figure 2.1.

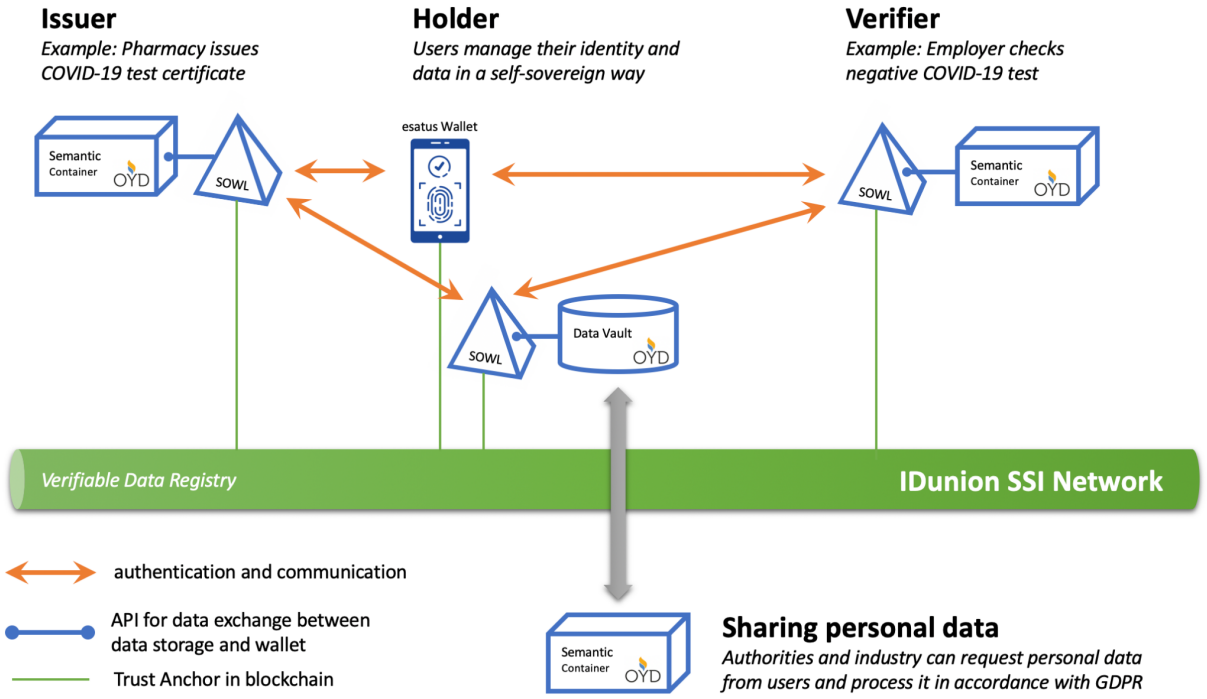

Figure 2.1: Overview

2.1 Semantic Container

Semantic Container are extended with the following functionality:

- integrate with SOWL: supporting another wallet besides TDA from Human Colossus Foundation
  - allow password-less login using OpenID Connect
  - store references to Verifiable Credentials and Verifiable Presentations managed in SOWL
- rendering for SOYA: replacing OCA as data capture mechanism
  - implement API endpoints and frontend application (DataBud) for data capture and validation
- visualising provenance: making information better accessible
  - using PlantUML to display Prov-O statements

2.2 Data Vault

The OwnYourData PDS is extended with the following functionality:

- storing consent information: upgrading to the current version of the Data Privacy Vocabulary
    - Version 0.4 was used as published here: https://github.com/w3c/dpv/tree/dpv-0.4
- managing credentials: allow to store meta-information about credentials that are managed in an SSI wallet
    - the current version of SOWL with a pre-configured tenant from esatus AG is used
- visualising provenance: making information better accessible (same as in Semantic Container)

2.3 Decentralised Identifier (OYDID)

The goal of this project was to also test the use of a decentralized identifier (DID) that is not based on a distributed ledger. In IDunion an Indy ledger (Sovrin) based identity network is established and provides the trust anchor for handling sensitive data. For certain aspects however, it could be interesting to have DIDs that don't require the full stack of a decentralized system. OYDID provides such a self-sustained environment for managing digital identifiers (DIDs). The did:oyd method links the identifier cryptographically to the DID Document and through also cryptographically linked provenance information in a public log it ensures resolving to the latest valid version of the DID Document.

In the course of the project a W3C conform DID Method Specification was developed and is available here: https://ownyourdata.github.io/oydid/

2.4 Semantic Overlay Architecture (SOyA)

SOyA allows data structures to be described in simple terminology. This description includes groups of data records with the same attributes, references between data records, and meta-attributes of these data structures.

Datasets are referred to as "bases", while meta-attributes are summarised in so-called "overlays". Overlays can contain information about attributes (e.g. detailed descriptions, permissible values, or formatting), but also transformations into other data structures.

For the exchange of these definitions of data structures, those structures can be stored in online repositories. When saving in such repositories, 2 versions are created:

- the original variant with the given names of the individual artefacts (i.e. of Structure, Base, and Overlay)
- a "frozen" variant where each name is replaced by a DRI; the DRI is a content-based address, which is also the unchangeable fingerprint of the content. This ensures that the previous version remains available if changes are made later.

With the described definitions of data structures, concrete data can now be recorded. SOyA offers the following functions:

- Acquire: If the attributes are named accordingly in a simple JSON ("flat JSON"), a conversion into JSON-LD can take place automatically
- Validate: Existing data records can be checked for conformity using a Validate overlay
- Transform: you can switch between data structures with a transformation overlay; It is thus possible to retain existing data formats (legacy formats), while automatic mapping to new standards is guaranteed
- Capture: with automatically generated HTML forms based on the structure information, data can be conveniently visualised, recorded and processed

In the course of the project a W3C conform Community Group Specification was developed and is available here: https://ownyourdata.github.io/soya/

2.4.1 Transformation between Covid Credentials

As base data structure for a Covid19 credential the WHO definition[1] was used:

WhoCovid19Credential	
header	→ Person
vaccinationEvent	→ VaccinationEvent
healthCertificateIdentifier	→ Identifier
certificateIssuer	string
certificateValidFrom	date
certificateValidUntil	date
certificateSchemaVersion	string
Person	
name	string
dateOfBirth	date
uniqueIdentifier	→ Identifier
sex	string
VaccinationEvent	
vaccineOrProphylaxis	string

[1] Digital Documentation of COVID-19 Certificates: Vaccination Status, Technical Specifications and Implementation Guidance from July 2021

vaccineBrand	string
vaccineManufacture	string
vaccineMarketAuthorization	string
vaccineBatchNumber	string
dateOfVaccination	date
doseNumber	integer
validFrom	date
totalDoses	integer
countryOfVaccination	string
administeringCentre	string
healthWorkerSignature	base64binary
healthWorkerIdentifier	→ Identifier
diseaseTarget	string
nextDoseDue	date
Identifier	
id	string
idSystem	string

The SOyA structure that describes the above definition is available here:

- https://soya.ownyourdata.eu/WhoCovid19Credential
- https://soya.ownyourdata.eu/zQmSREDaikBfZ2fPFVfSaFbXicNYDkAjmkbSumwfQXNmAs4

The WHO Covid19 definition is almost identical to the EU Green Pass definition and therefore the formats of the IATA Travel Pass and the Good Health Pass initiative were evaluated for transformation.

IataTravelPass		WhoCovid19Credential mapping
vaccinationEvent	→ vaccinationEvent	n.a.
dateOfVaccination	date	vaccinationEvent > dateOfVaccination
doseNumber	integer	vaccinationEvent > doseNumber
countryOfVaccination	string	vaccinationEvent > countryOfVaccination

administeringCentre	string	vaccinationEvent > administeringCentre
vaccineBatch	string	vaccinationEvent > vaccineBatchNumber
nextDoseDue	date	vaccinationEvent > nextDoseDue
VaccinationEvent		
personIdentification	→ Person	n.a.
vaccine	string	vaccinationEvent > vaccineOrProphylaxis
brand	string	vaccinationEvent > vaccineBrand
diseaseTarget	string	vaccinationEvent > diseaseTarget
Person		
name	string	header > name
dateOfBirth	date	header > dateOfBirth
uniqueIdentifier	string	header > uniqueIdentifier > id
sex	string	header > sex

The SOyA structure that describes the above definition together with a transformation overlay from WHO to IATA Travel Pass is available here:

- https://soya.ownyourdata.eu/IataTravelPass
- https://soya.ownyourdata.eu/zQme9BFM1Luar19rMNNCWpoTSUheoLKrkc1VPWQWGB2yXNy

GoodHealthPass		
recipient	Person	n.a.
disease	string	vaccinationEvent > diseaseTarget
vaccine	string	vaccinationEvent > vaccineOrProphylaxis
productName	string	vaccinationEvent > vaccineBrand
cvxCode	string	-
marketingAuthorizationHolder	string	vaccinationEvent > vaccineMarketAuthorization
doseNumber	integer	vaccinationEvent > doseNumber
dosesPerCycle	integer	vaccinationEvent > totalDoses
dateOfVaccination	date	vaccinationEvent > dateOfVaccination

stateOfVaccination	string	-
countryOfVaccination	string	vaccinationEvent > countryOfVaccination
certificateIssuer	string	certificateIssuer
certificateNumber	string	healthCertificateIdentifier > id
Person		
givenName	string	header > name (part: first name)
middleName	string	header > name (part: middle name)
familyName	string	header > name (part: family name)
birthDate	date	header > dateOfBirth
photoy	binary	-

The SOyA structure that describes the above definition together with a transformation overlay from WHO to Good Health Pass is available here:

- https://soya.ownyourdata.eu/GoodHealthPass
- https://soya.ownyourdata.eu/zQmZg42np1LXN9pCxJfoMPBiDBaQMdFt7A3ZrsxrQ2fFDRs

2.5 Consent Management

According to the SPECIAL project[2], "*a Usage Policy is meant to specify a set of authorised operations*". Within a Semantic Container and the Data Vault, such authorised operations document consent and are characterised by mandatory attributes used to check compliance between a Data Subject and a Data Controller, and informational attributes to document additional aspects.

Mandatory attributes from the Data Privacy Vocabulary:

- *Data Categories:* the type of data processed by the operation
  `dpv:hasPersonalData`
- *Purpose:* the purpose of the operations by the Data Controller
  `dpv:hasPurpose`
- *Processing:* a list of operations that can be executed by the Data Controller
  `dpv:hasProcessing`
- *Recipient:* the entities that can access the results of the operations
  `dpv:hasRecipient`
- *Location:* where data can be stored
  `dpv:hasLocation`

[2] https://www.specialprivacy.eu

- *Duration:* how long the data can be stored
  `dpv:hasExpiryTime`
- *Storage:* a description of
  - *Location:* where the result is stored and
  - *Duration:* for how long

Attributes with informal character from the Data Privacy Vocabulary:

- *Technical Measures:* technical measures applied when handling the data
  `dpv:hasTechnicalMeasure`
- *Organisational Measures:* organisational measures applied when handling the data
  `dpv:hasOrganisationalMeasure`
- *Legal Basis:* legislations and regulations applicable to the dataset
  `dpv:hasLegalBasis`
- *Risk:* associated risks or possibilities of negative effects, impacts, or consequences
  `dpv:hasRisk`

Stakeholders using a Semantic Container or the OwnYourData Data Vault can specify their own Usage Policy and upon exchange between those parties compliance is automatically verified and documented. The process of evaluating compliance between Usage Policies from 2 parties uses a reasoner (rules engine) that evaluates the RDF graphs from Data Subject and Data Controller.

2.6 Provenance

The Provenance information in a Semantic Container and Data Vault provides an audit trail for the data stored. Based on to the PROV-Ontology (https://www.w3.org/TR/prov-o/), the following three classes are used to specify provenance:

- **Entity:** used to describe data stored in a Semantic Container / Data Vault
- **Agent:** provides information about the storage capabilities
- **Activity:** provides information about the data source and/or processing performed

Using PlantUML provenance information in a record is visualised.

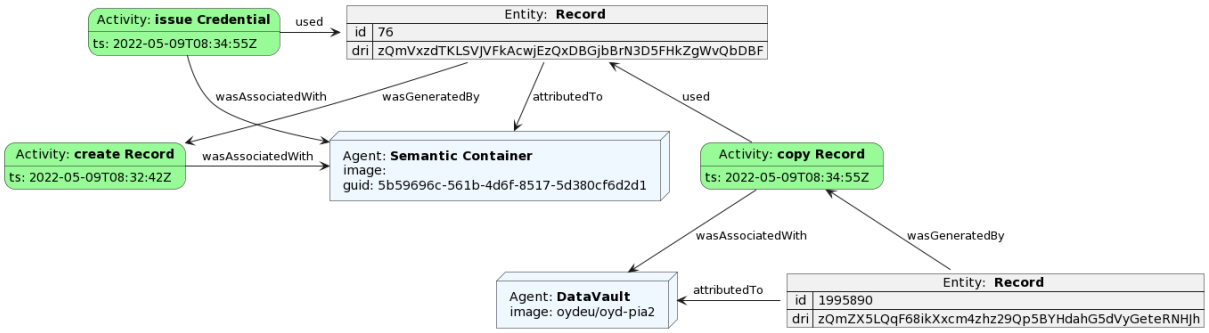

Figure 2.2: Example provenance visualisation

3 Interfaces and workflows

This section describes interfaces of the components listed in chapter 2 and details the workflows for the IDunion project.

3.1 Data structures

A focus of this project is to provide a reference implementation to exchange COVID credentials and other relevant associated data between different standards and formats. The following types of information are covered:

- Doctor information
- research information
- Usage Policies (according to DPV v0.3)
- Vaccine information
- Verifiable Credential for Vaccination (COVID credential)
- Verifier information

3.2 Creating a Verifiable Credential

The first dataflow describes the necessary steps for a user to acquire a verifiable credential for vaccination and necessary infrastructure on the issuer side to provide such a credential.

3.2.1 Preparation

This section describes the necessary steps to set up the infrastructure for user and issuer. Figure 3.1 depicts the sequence diagram and it includes 4 possible scenarios for users:

1. no initial infrastructure available
2. PDS account available
3. SSI wallet available
4. PDS account and SSI wallet available

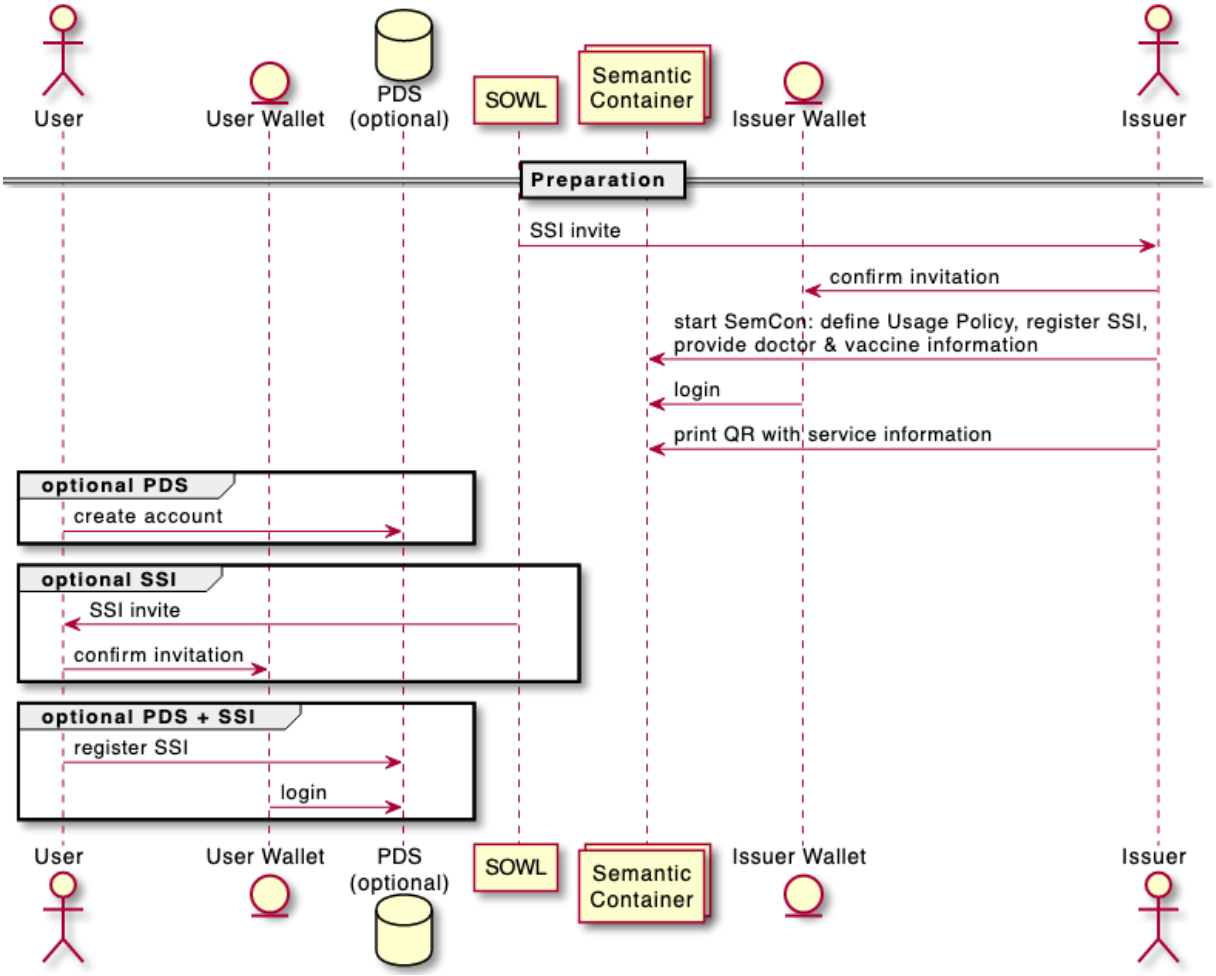

Figure 3.1: Prepare creating verifiable credential

3.2.2 Gather information

This section only applies for users with a PDS account and describes the process to review information about issuer (doctor) and vaccination.

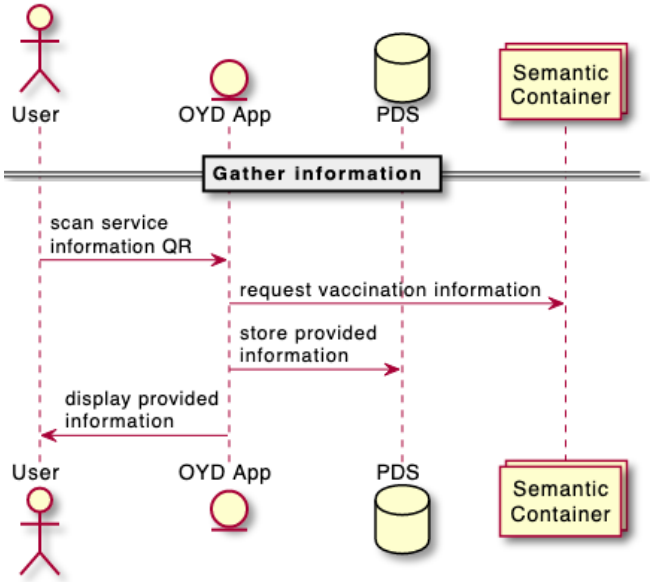

Figure 3.2: Gather information about vaccination

3.2.3 Create Verifiable Credential

This section describes the process of requesting and creating a verifiable credential.

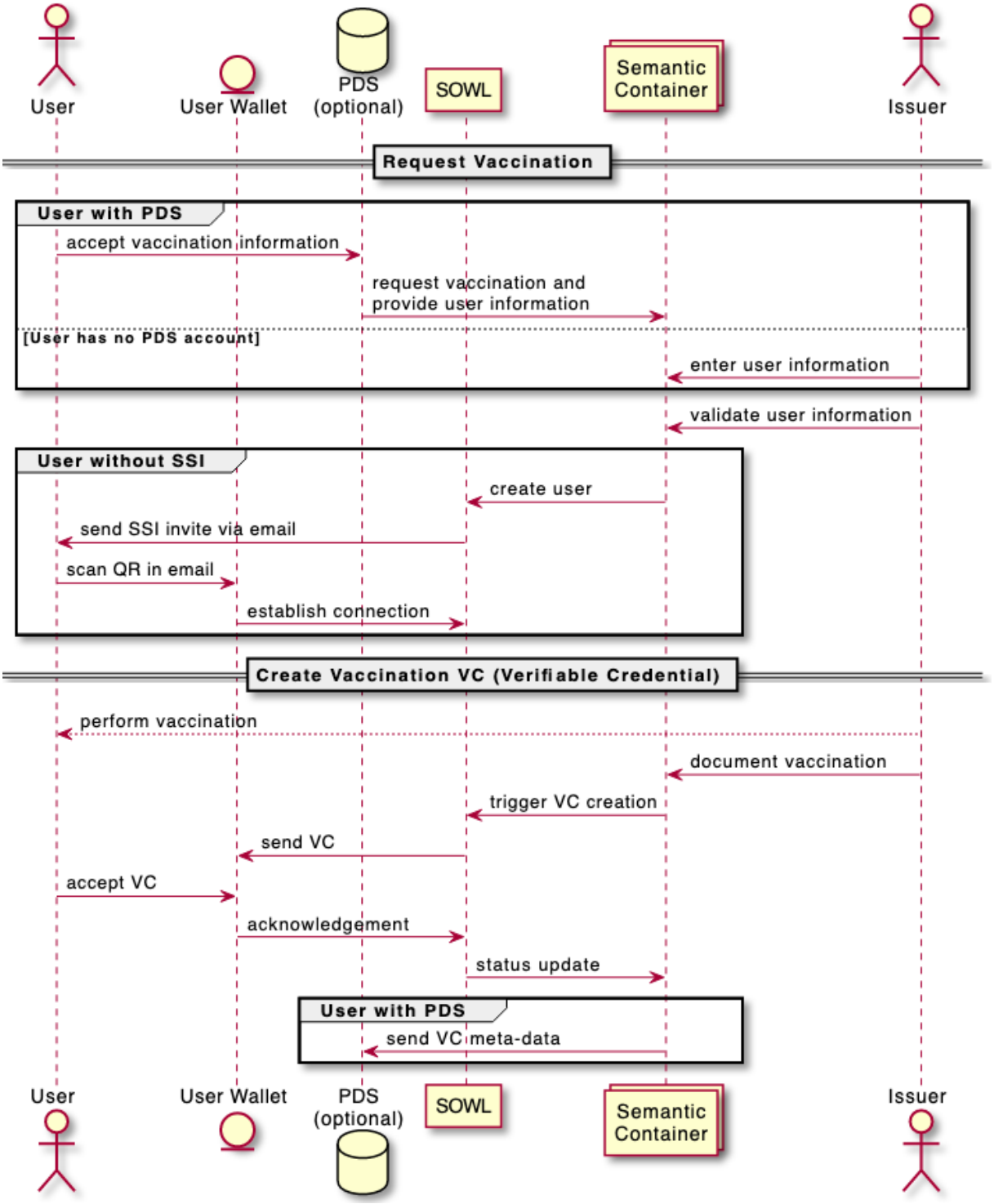

Figure 3.3: Create verifiable credential

3.3 Verifying a Credential

This data flow describes the steps to prove immunisation status to a verifier.

3.3.1 Preparation

This section describes a user approaching a checkpoint and gathering information about the organisation requesting a credential.

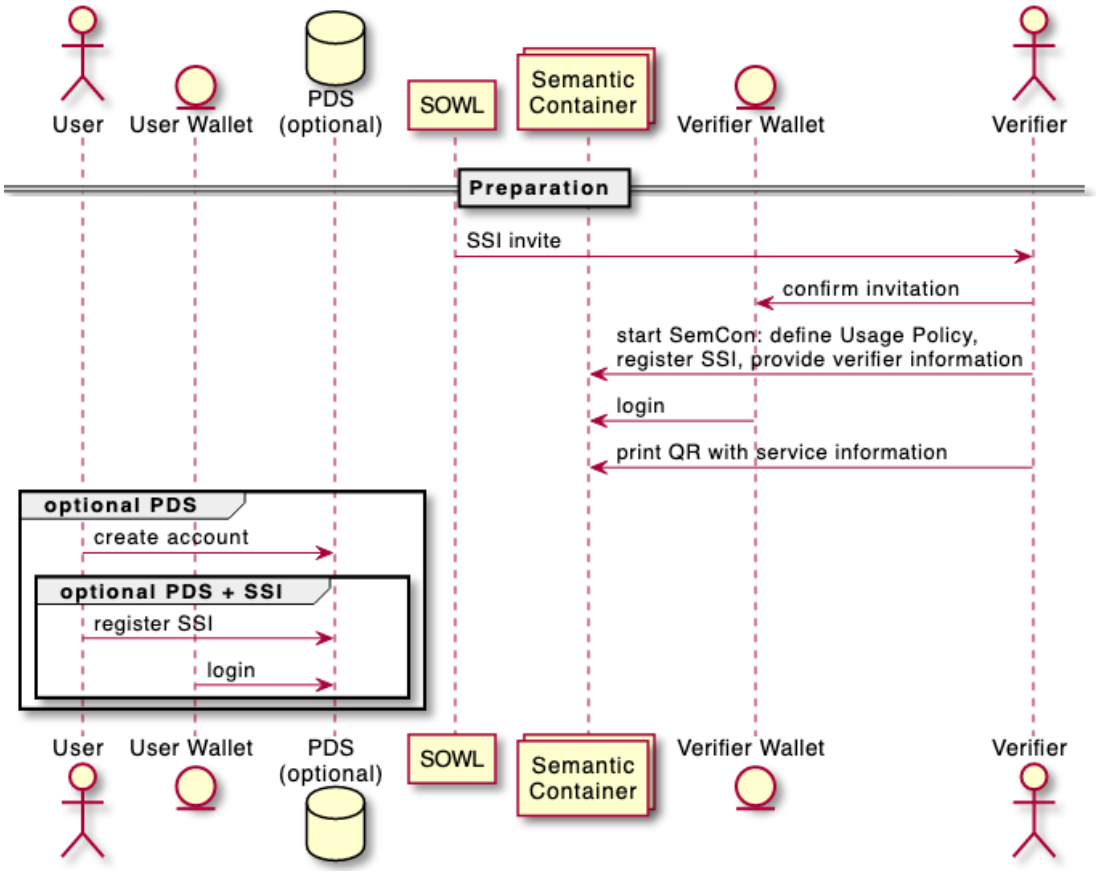

Figure 3.4: Prepare proofing immunization status

3.3.2 Gather information

This section only applies for users with a PDS account and describes the process to review information about verifiers.

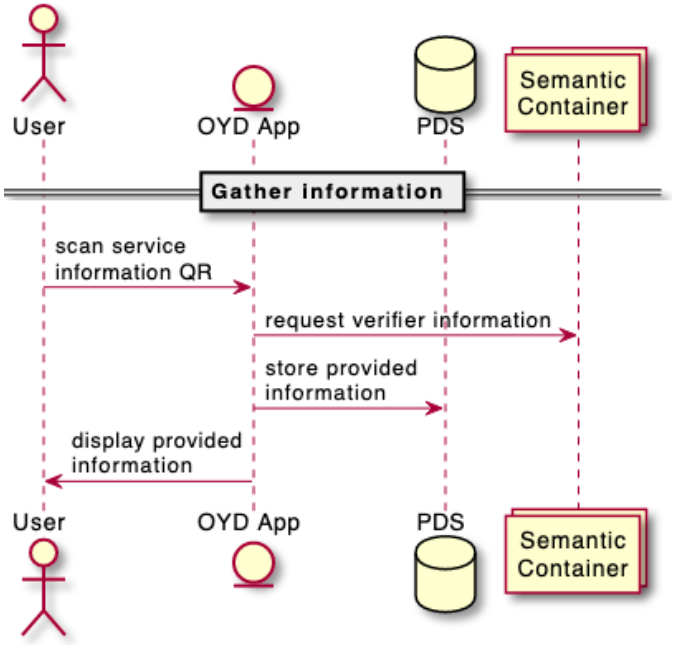

Figure 3.5: Gather information about verifier

3.3.3 Verify credential

This section describes a user proofing the immunizations status and a verifier approving or denying the presented credential.

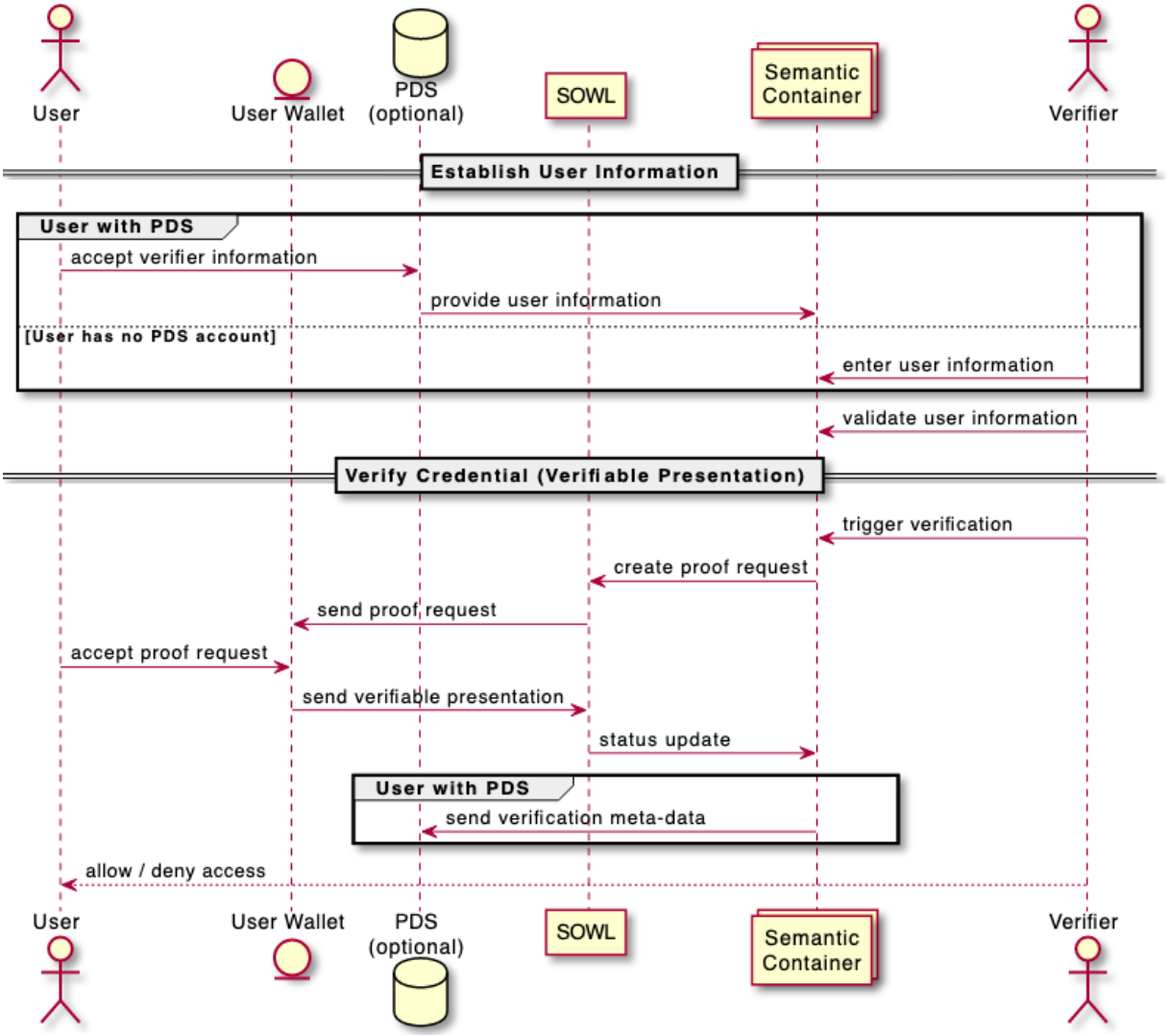

Figure 3.6: Present information

3.4 Sharing personal data with an organisation

The final workflow describes the privacy-preserving sharing of personal data with a 3rd party.

3.4.1 Contact participants

This section describes a research institution inviting users to share data.

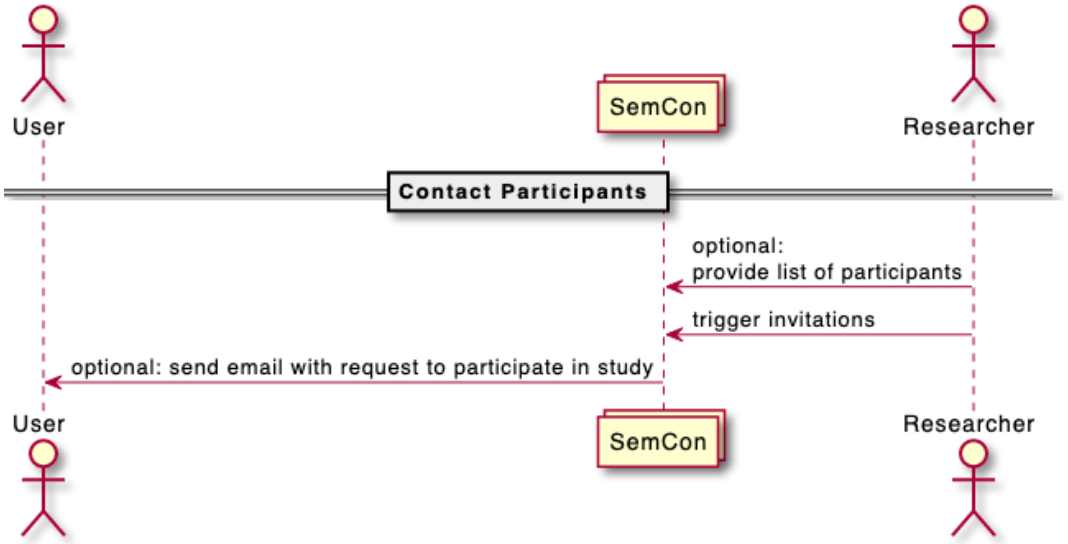

Figure 3.7: Contact participants

3.4.2 Participate in data sharing

This section describes a user reviewing information about the data sharing request, sending data to share, and storing information about sent data.

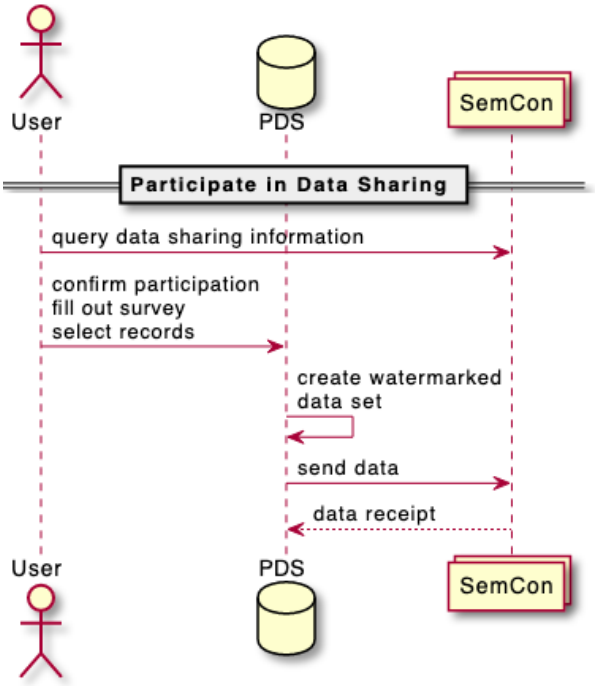

Figure 3.8: Participate in data sharing

3.4.3 Data tracing

This section describes a user requesting information about the shared data and revoking consent for any further use of the data.

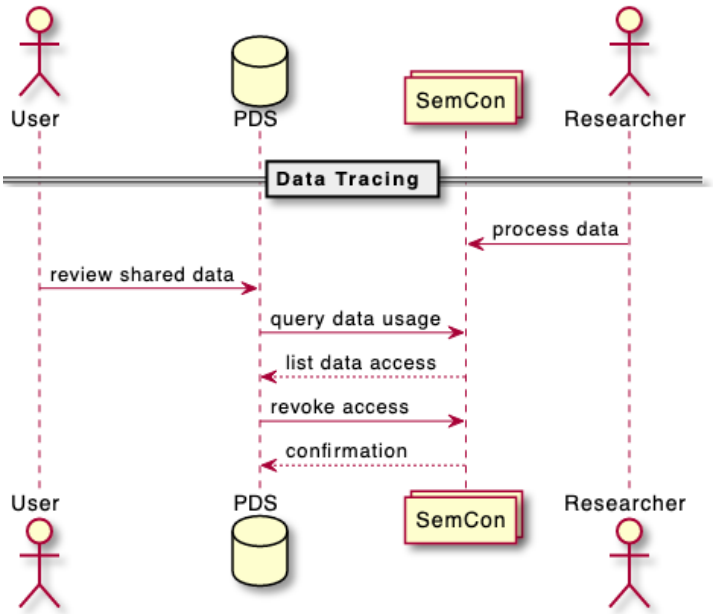

Figure 3.9: Trace data

4 System testing

This chapter describes scope, setup, and results of the demonstrator developed in the course of the project. Based on the requirements described in D2.1 the functionality was implemented and tested as described below.

4.1 Test cases

To demonstrate the functionality of the system the following end-to-end test cases are used for verification.

1) Creating a Verifiable Credential

   Tests:

   a) Simple Flow: with esatus wallet only
   b) Complex Flow: with Data Vault account and esatus wallet

   Simple Flow:

   1. start Semantic Container with Issuer functionality configured
   2. click "New Vaccination" in "Schemas" view and fill out presented form
   3. click "Vaccinate" to send Credential Offer to esatus Wallet
   4. confirm Credential in esatus Wallet
   5. trigger scheduler in Semantic Container (upper right corner) to archive the record

   Complex Flow:

   Semantic Container - Issuer

   1. start Semantic Container with Issuer functionality configured https://idunion-issuer.data-container.net
   2. use "Login with SOWL" to perform OpenID-Connect login method
   3. print QR code for users (upper right corner)

   OwnYourData App - User

   4. launch OwnYourData App and choose "QR Connect" extension
   5. scan QR code from Semantic Container
   6. review Issuer information and Usage Policy compliance
   7. confirm data exchange by clicking "Send"

   Semantic Container - Issuer

   8. incoming vaccination requests are shown with the Schema "IDuVacCert_Application" and are reviewed by the doctor (issuer)
   9. click "Vaccinate" to send Credential Offer to esatus Wallet
   10. confirm Credential in esatus Wallet
   11. trigger scheduler in Semantic Container (upper right corner) to archive the record and share data with user

   OwnYourData App - User

   12. show vaccination record and metadata in Data Vault using Data Bud extension

2) Proofing Immunisation Status

Prerequisites:

- OwnYourData App and esatus Wallet installed on smartphone
- Semantic Container for verifier deployed
- SOWL configured

Tests:

a) Simple Flow: with esatus wallet only
b) Complex Flow: with Data Vault account and esatus wallet

Simple Flow:

1. start Semantic Container with Verifier functionality configured
2. click "New Vaccination Proof" in "Schemas" view and fill out presented form (fields are pre-filled with first record of schema "IDuProof_Template")
3. click "Request" to send Proof Request to esatus Wallet
4. confirm Proof in esatus Wallet
5. trigger scheduler in Semantic Container (upper right corner) to archive the record

Complex Flow:

Semantic Container - Issuer

1. start Semantic Container with Verifier functionality configured https://idunion-verifier.data-container.net
2. use "Login with SOWL" to perform OpenID-Connect login method
3. print QR code for users (upper right corner)

OwnYourData App - User

4. launch OwnYourData App and choose "QR Connect" extension
5. scan QR code from Semantic Container
6. review Verifier information and Usage Policy compliance
7. confirm data exchange by clicking "Send"

Semantic Container - Issuer

8. incoming data is shown with the Schema "IDuProof_Application" and is reviewed by officer (verifier)
9. click "Request" to proof Vaccination Status using esatus Wallet
10. confirm Proof Request in esatus Wallet
11. trigger scheduler in Semantic Container (upper right corner) to query response from user wallet
12. data is shown with the Schema "IDuProof_Processed" and officers maks final decision about acceptance / rejection
13. archive the record and share data with user

OwnYourData App - User

14. show checkpoint record and metadata in Data Vault using Data Bud extension

3) Sharing Data with 3rd party

Prerequisites:

- OYD Data Vault with personal information
- Semantic Container from organisation for data sharing setup

Tests:

a) Setup
   1. OYD Data Vault Account and Data Sharing Plugin installed
   2. Semantic Container configured to receive shared data
   3. defined Usage Policy for receiving data
   4. list of contacts participating in data sharing

b) Organisations sends invitation to data sharing
   5. send email to planned participants with DID for further information
   6. retrieve and show information from Semantic Container in OYD Data Vault
   7. select data to be shared

c) Send data
   8. apply digital watermarking to selected data
   9. send watermarked data together with Usage Policy to Semantic Container
   10. store information about shard data

d) Manage shared data
   11. retrieve information about shared data
   12. revoke consent for shared data
   13. identify if a dataset is own dataset (based on digital watermark) and whom it was shared with

4.2 Test results

In the course of the project a number of end-to-end tests are performed to track development and ensure successful integration of the numerous components. The following graphic outlines the test system setup.

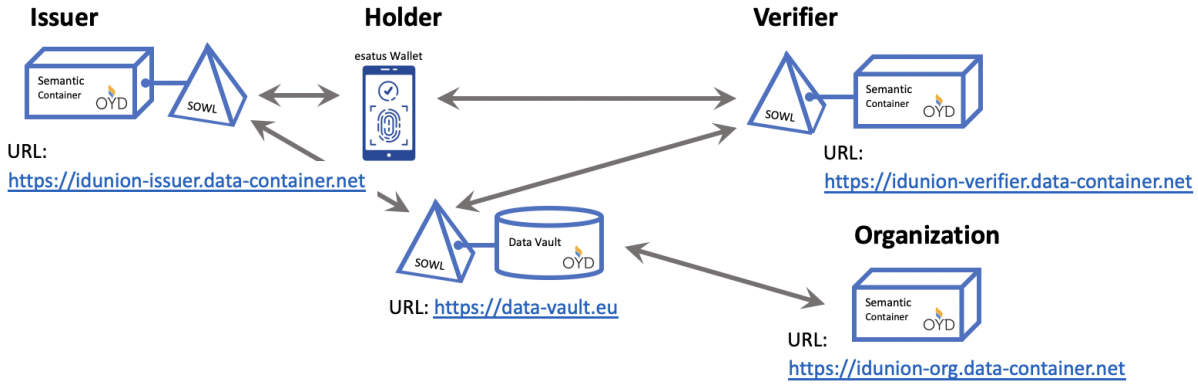

Figure 4.1: IDunion Test System Setup

Status of system for IDunion tests:

- Semantic Container / DataBud v0.15.32
- esatus Wallet v1.1 (Build 11108)
- SOWL v1.8
- Data Vault v0.8.2
- OwnYourData App v0.6.0
- OYDID v0.4.7
- SOyA v0.15.2

1) Creating a Verifiable Credential: Complex Flow (with Data Vault account and esatus wallet)

   Prerequisites:

   - OwnYourData App and esatus Wallet installed on smartphone
   - Semantic Container for issuer deployed: https://idunion-issuer.data-container.net
   - SOWL configured
     - Tenant ID: 17934
     - Application ID (OIDC): 23183
     - Credential Definition ID: 42745

   Tests:

   Semantic Container - Issuer

   1. start Semantic Container with Issuer functionality configured
      - APP-KEY: `7509b6d86591c4e354c649e6b9bde323cd9fab9c456727c2f3ba2960efd309ba`
      - APP-SECRET: `097a8c0beb0365309460ec7b368a34d139f74818bfc12143019286ef86cb6a97`
   2. use "Login with SOWL" to perform OpenID-Connect login method
   3. print QR code for users (upper right corner)
      DID: `did:oyd:zQmTKdtV2beKbJxTjWLdBqVe8UY1ddms7xSLq4ePHC2uNuc`

   OwnYourData App - User

   4. launch OwnYourData App and choose "QR Connect" extension
   5. scan QR code from Semantic Container
   6. review Issuer information and Usage Policy compliance
      - Usage Policies don't match because of Expiry Date
   7. confirm data exchange by clicking "Send"

   Semantic Container - Issuer

   8. incoming vaccination requests are shown with the Schema "IDuVacCert_Application"
      - values filled with information provided by user
      - other fields are pre-filled with values from "IDuVacCert_Template"
   9. Doctor (Issuer) reviews input and fills out relevant fields
   10. click "Vaccinate" to send Credential Offer to esatus Wallet
   11. user confirms Credential in esatus Wallet
   12. trigger scheduler in Semantic Container (upper right corner) to archive the record and share data with user
       - Vaccination data is shown in "IDuVacCert_Archived"

OwnYourData App - User

13. show vaccination record and metadata in Data Vault using Data Bud extension

Verification:

- Treatment is documented in Semantic Container
- Verifiable Credential is listed in Personal Data section
- complete provenance visualisation in Data Vault

2) Proofing Immunisation Status: Complex Flow (with Data Vault account and esatus wallet)

Prerequisites:

- OwnYourData App and esatus Wallet installed on smartphone
- Semantic Container for verifier deployed: https://idunion-verifier.data-container.net
- SOWL configured
    - Tenant ID: 17934
    - Application ID (OIDC): 23183
    - Proof Template ID: 47627

Tests:

Semantic Container - Verifier

1. start Semantic Container with Verifier functionality configured https://idunion-verifier.data-container.net
    - APP-KEY: `7d7bd106ee8c00e56e70798110ed1683f4088110928cfe8a2e49067260c0c9b1`
    - APP-SECRET: `d52f4320698a4e4f959a2fa4e2d07ba58695c8ae2e284cc235d7179773810bfb`
2. use "Login with SOWL" to perform OpenID-Connect login method
3. print QR code for users (upper right corner)
   DID: `did:oyd:zQmY1xpL1KLqYYowqPWD1K61RguMxHz34eNoXWgK5kS5qFc`

OwnYourData App - User

4. launch OwnYourData App and choose "QR Connect" extension
5. scan QR code from Semantic Container
6. review Verifier information and Usage Policy compliance
    - Usage Policies don't match because no compliance in dpv:hasProcessing
7. confirm data exchange by clicking "Send"

Semantic Container - Verifier

8. incoming data is shown with the Schema "IDuProof_Application" and is reviewed by officer (verifier)
9. click "Request" to proof Vaccination Status using esatus Wallet
10. confirm Proof Request in esatus Wallet
11. trigger scheduler in Semantic Container (upper right corner) to query response from user wallet
12. data is shown with the Schema "IDuProof_Processed" and officers maks final decision about acceptance / rejection
13. archive the record and share data with user

OwnYourData App - User

14. show checkpoint record and metadata in Data Vault using Data Bud extension

Verification:

- Verification is documented in Semantic Container
- Verifier and verification information is stored in OYD Data Vault
- complete provenance visualisation in Data Vault

3) Sharing Data with 3rd party

Prerequisites:

- OYD Data Vault with personal information
  URL: https://data-vault.eu
  user: idunion@ownyourdata.eu
  password: Secr3t!
- Semantic Container from organisation for data sharing setup
  https://idunion-org.data-container.net
  - APP-KEY:
    `cb043019489112253734e484c1ae24d6926b65e9224d1beeb46211d9b20faa09`
  - APP-SECRET:
    `06364771e9dfc9f355c1c8ae2e12d2f9e90cd814fd2f5925f808e3f93af88c35`

Tests:

Setup

1. OYD Data Vault Account and Data Sharing Plugin installed
2. Semantic Container configured to receive shared data
3. defined Usage Policy for receiving data
4. list of contacts participating in data sharing

Organisations sends invitation to data sharing

5. send email to planned participants with DID for further information
6. retrieve and show information from Semantic Container in OYD Data Vault
7. select data to be shared

Send data

8. apply digital watermarking to selected data
9. send watermarked data together with Usage Policy to Semantic Container
10. store information about shard data

Manage shared data

11. retrieve information about shared data
12. revoke consent for shared data
13. identify if a dataset is own dataset (based on digital watermark) and whom it was shared with

Verification:

- list of retrieved data (dataset) for organisation
- information about data sharing in OYD Data Vault

5 Conclusions

5.1 Software repositories

The various components described in this document are all open source and available on Github as listed in the following table.

Component	Github Repository
PDS (Data Vault)	https://github.com/OwnYourData/oyd-pia2
Semantic Container (base image)	https://github.com/sem-con/sc-base
IDunion specific Semantic Container	https://github.com/OwnYourData/sc-idunion
DataBud (access records in PDS or SemCon)	https://github.com/OwnYourData/oyd-databud
did:oyd Methode (OYDID)	https://github.com/OwnYourData/oydid
Semantic Overlay Architecture (SOyA)	https://github.com/OwnYourData/soya
DPV Service (reasoner to match Usage Policies)	https://github.com/OwnYourData/dpv-service
OYD Form (scan QR code for SemCon connection)	https://github.com/OwnYourData/oyd-form

5.2 Outlook

In this project we demonstrate the secure and private exchange of credentials between organisations and individuals as well as individuals sharing data with organisations. With technologies like decentralised identifiers and Personal Data Stores a new way of managing personal data emerges that can also be utilised in other areas like supply chain management. With recent legislation of Digital Product Passports a new field for deploying this technology is created.

Appendix

Glossary

Below is a list of acronyms and abbreviations used throughout the document.

Acronym	Description
DID	Decentralised Identifier
PDS	Personal Data Store (in this project: OwnYourData Data Vault)
VC	Verifiable Credential
SPECIAL	Scalable Policy-aware Linked Data Architecture For Privacy, Transparency and Compliance (project website: https://www.specialprivacy.eu/)