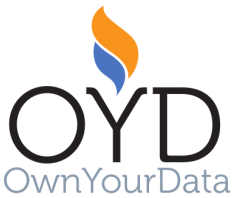
Horizon 2020 Programme
DG CNECT
Next-generation Internet

NGI DAPSI

Project acronym: **DIP**

Project title: **Digital Immunization Passport**



HUMAN COLOSSUS
FOUNDATION

Deliverable 2.2
Design Document

Deliverables leader:	**OwnYourData**
Authors:	Christoph Fabianek, Eduard Gringinger, Gabriel Unterholzer, Philippe Page, Paul Knowles, Robert Mitwicki, Meri Seistola
Due date:	2020-11-20
Actual submission date:	2020-11-20
Dissemination level:	Public

Abstract: This report is part of a third-party project DIP that has received funding from the NGI_DAPSI (DAPSI 1st open call), the European Union's Horizon 2020 research and innovation programme under grant agreement No. 871498.

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

Document Revision History

Date	Version	Author/Editor/Contributor	Summary of main changes / Status
2020-09-07	0.1	Christoph Fabianek	Initial document
2020-11-20	1.0	Christoph Fabianek	First public version
2021-01-29	1.1	Christoph Fabianek	updates towards 2nd evaluation
2021-03-22	1.2	Christoph Fabianek	Demo System and extended tests
2021-05-31	1.3	Christoph Fabianek	load tests, final updates

Disclaimer

Copyright

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

Table of content

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

Executive Summary

This deliverable provides the design description of the Digital Immunization Passport (DIP) 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 emphasizes the use of an upcoming standard for data exchange in Personal Data Stores (i.e., CEPS - Common Endpoints for Personal data Stores). It continues with 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 an extensive test procedure 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 NGI DAPSI.

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

1 Introduction

1.1 Background

Currently, vaccination and immunization information are spread over different organizations like labs and hospitals as well as pharmaceutical companies together with government agencies. A patient usually only has a paper certificate that provides vaccination treatments with often difficult to read handwritten additional information.

The main focus of this project is on Data Interoperability & Compatibility through establishing interfaces between the health industry and individuals as well as pushing forward on standardized 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 DIP 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 organizations
- D2.2 Design Specification: describes and depicts the system design together with API endpoints and data formats of the various components

FUNDED BY NGI 
This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

2 Functional Components

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

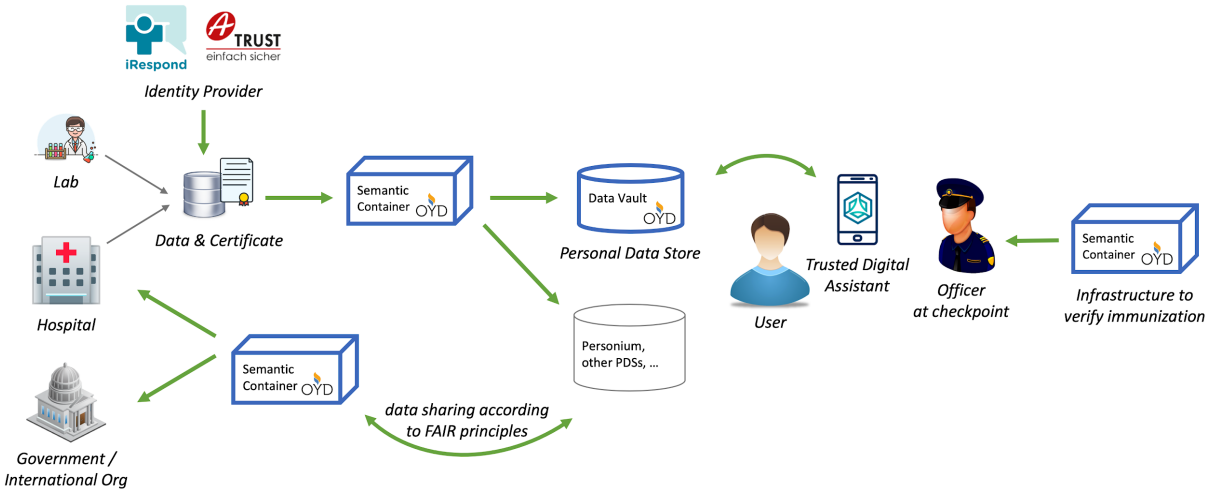


Figure 2.1: Overview

2.1 Personal Data Store

A general introduction to the OwnYourData Data Vault (the PDS [Personal Data Store] used in this project) is available in Deliverable 2.1 Requirements Specification in section 3.2.1. The development and further extension of the Data Vault in the course of this project focused on the following areas:

- Create unified endpoints (APIs) to allow full data portability between other PDSs
- Implement a plugin to process invitations to participate in a data sharing request
- provide digital watermarking for sharing data with 3rd parties

2.1.1 Common Endpoints

The OwnYourData Data Vault is a client-server application that exposes all server-side functionality through a well documented API. Within MyData a group of PDS developers joined to discuss possibilities to standardize basic operations to access a PDS and this was published as CEPS v1.0. Currently, CEPS v2.0 is in the works and there will be a focus on data sharing not only between PDSs but also with 3rd parties. A dedicated data flow in this project is therefore the exchange of data between individuals and institutions without any monetary compensation - sometimes also referred to as "Data Altruism".

Section 3.1 Data Structure & API describes in detail the new data structure that was developed to read and write data and cater for the different needs of various

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

endpoints: requesting the actual data payload (content), metadata, and/or validation information. Additionally, the set of API endpoints is described there that was built based on the learnings from CEPS v1.0 but also includes relevant advancements based on the use of schema information provided through OCA.

The practical test for the endpoints was the implementation of the DataBud (see section 2.3.2) as well as the Data Sharing plugin described in the next section.

2.1.2 Data Sharing

Sharing data in a secure and privacy preserving manner requires a number of preconditions and some of those can be fulfilled by technical measures. In the course of this project the following privacy preserving mechanisms in the data sharing dataflow will be used:

- *proofing own rights about shared data:* through the means of Data Provenance (using the PROV Ontology[1]) a user can document the origin of his or her data and together with storing a hash value of this information in a distributed ledger (blockchain) the data become immutable (i.e., a defined state at a given point in time)
- *specifying usage policies for shared data:* by attaching machine readable information about the allowed use for a given dataset (using the Usage Policy Language[2]) a data subject and a data controller can automatically verify compliance
- *tracing data:* any data recipient provides a unique token that can be used to query what data was received and if it was handed over to subsequent recipients
- *revoking access:* in case a user wants to revoke consent for a shared dataset the recipient must provide mechanisms to remove the data and also recursively propagate such a revocation request to any subsequent recipients of this data
- *identifying leaks:* in case a negligent or malicious recipient leaks received data (i.e., uses data beyond the scope defined in the usage policy) a user is able to identify this data and also identify who was the original recipient of the data - the technology of digital watermarking to identify leaks is described in the next section

2.1.3 Digital Watermarking

Embedding a watermark into a dataset includes a number of steps:

- *Preprocessing:* a dataset is partitioned into fragments with a maximal size and it is assured that a numerical value exists in each record; into this numerical value the digital watermark will be embedded
- *Error vector:* based on the user and usability requirements for the data a unique error vector is generated and stored

[1] https://www.w3.org/TR/prov-o/
[2] https://www.specialprivacy.eu/images/documents/SPECIAL_D21_M12_V10.pdf

FUNDED BY NGI
This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

- *Encoding:* the error vector is added to the numerical value field and generates the watermarked fragment

With the approach above a number of possible attacks exists to disturb the embedded watermark:

- *Distortion attack:* rounding to the n-th digit in the numerical value field
- *Deletion attack:* remove a subset of the records in a dataset
- *Mean collusion attack:* combine n copies of the same dataset with different watermarks applied and replace the numerical value with the mean of n values
- *Combined attack:* use a combination of the attacks described above

The planned implementation is already robust against distortion and deletion attacks but further work is necessary on other attacks.

Verifying if a suspicious dataset is a copy of own data and answering the question on who was the recipient of this dataset, is performed with the following steps:

- *Preprocessing:* fragmenting the suspicious dataset based on the original partitioning process (e.g., a fragment holds data from one month)
- *Detect original data:* compare each watermarked suspicious fragment with own data and calculate the probability that it is a match for a given fragment
- *Identify recipient:* compare original fragments with stored error vectors applied and calculate similarity with the suspicious fragment

Note: the technology of applying Digital Watermarks was first developed for Semantic Containers in the course of the EU funded NGI_TRUST project "MyPCH" and this project extends the technology to Personal Data Stores.

2.2 Trusted Digital Assistant (TDA)

(see D2.1 requirements section 3.2.3 for the definition of Trusted Digital Assistant)

The Digital Immunization Passport project requires interaction between multiple parties. To be able to secure those interactions, a TDA establishes a secure connection between two parties via the *DID auth* method. DID Auth uses the concept of DIDs with encryption mechanisms coming from pairwise keys generated for the purpose of that connection. To avoid correlation attacks each interaction can use completely random sets of DIDs. DID Authentication is based on simple challenge exchange which proves ownership of private keys behind the used DID. This way without knowing anything about other parties we can establish a secure communication channel which would allow us to start establishing trust by verifying verifiable information about the other party. Below the diagram shows how such an interaction looks like:

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

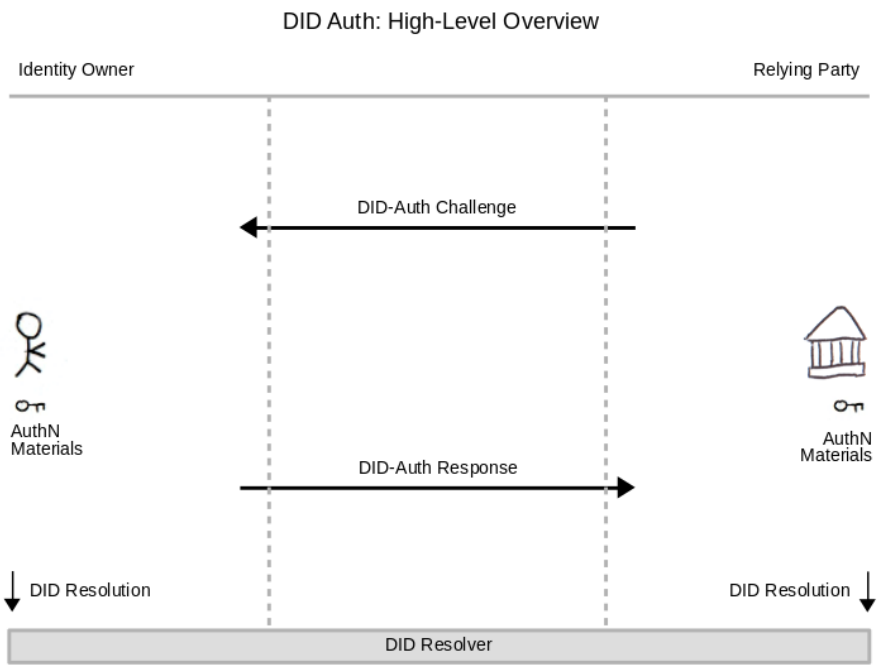


Figure 2.2: DID Auth

Having a secure communication channel allows both parties to start exchanging information. Information exchange is happening via the DIDComm protocol[3]. DIDComm is a high level protocol which is message-based, asynchronous, and simplex. Basically, it describes how two or more parties can communicate with each other establishing a security layer using only their own keys without any centralized authority.

One of the prominent use cases within the DIP project is data exchange between user and a clinic respectively a checkpoint. TDA leverages technologies like OCA, Semantic Container and PDS to be able to retrieve, store, and manage any type of data sets on behalf of the user. A user can configure a TDA to link it with data sources (SemCon, PDS, etc.) and through schema base decentralized resource identifier (DRI) it can manage available data sets. Each time a TDA is asked to provide data for issuing verifiable credentials, it reuses known sources by retrieving data based on a unique DRI.

Due to the user centric nature of a TDA any data exchange is attached with a user consent. Consent describes the conditions under which data is shared, e.g., how long data would be used, where they would be stored, if 3rd parties would have access to the data, etc. Users can define their own consent via a TDA interface and attach it to any data exchange. When exchanging data the TDA generates Verifiable Credentials which are issued to the party who receives the data as a

3 https://github.com/hyperledger/aries-rfcs/blob/master/concepts/0005-didcomm/README.md

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

proof that the user gave consent to use this data. This way all aspects of the data flow can be cryptographically secured and are auditable by the user or any external auditor. This provides a mechanism for the user to manage consent and also revoke it at any time.

"A" for assistant is an essential component of the TDA, not a marketing term. The TDA truely and solely acts for the benefit of the user. It shields the user from the complexities of managing his/her identity online by taking care of everything that is machine readable. Leaving the user with the only human level decision making (e.g. profiles, preferences).

The development path of the TDA goes well beyond the DIP project. Innovative UI, machine learning and AI algorithms are to be developed but this is outside the current scope. In the current DIP project, the TDA plays nevertheless a crucial role in managing:

- DIDs
- Invitations
- Connections
- Messages
- Documents
- Services
- Consents

on behalf of the user.

2.3 Semantic Container

The concept of Semantic Containers (SemCon) was introduced in Deliverable 2.1 Requirements Specification in section 3.2.4. In this project SemCon provides the platform to support functionalities to manage identities and share data. This addresses the fact that organizations sometimes struggle to deploy the complete stack of components to successfully use a new technology. Through containerization the complex dependencies are hidden behind a well documented API and allow quick deployment as well as easier maintenance.

In the Digital Immunization Passport project, SemCon creates a package containing all required functionality for the individual organizations:

- *Clinic:* provide an interface to the TDA representing doctors (credential issuers) and store data about all vaccinated users
- *Checkpoint:* provide an interface to the TDA representing officers (credential verifiers) and store data about all users that had their credentials checked
- *Research Institution:* maintain an optional participant list and send out invitation emails; provide an endpoint to accept data shared by individuals - section 2.3.3 gives a detailed description of this container

2.3.1 Common Endpoints

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

The same way as the standardized API endpoints for the PDS described in section 2.1.1 allow for interoperability between PDSs, the APIs on the organization side provide an interoperable way to exchange data with any PDS (this is demonstrated in the Data Sharing dataflow in section 3.4; communication between Clinic/Checkpoint and the user is performed through the TDA).

An additional benefit during the development and refinement of the API was the ability to provide developers with a single Docker container that behaves like a PDS. The OwnYourData Data Vault requires a number of components and also more resources to run, while a Semantic Container is light-weight and quick to start up.

2.3.2 DataBud

Early on in the project it turned out that a versatile record viewer and editor was necessary to access records in a Semantic Container and also make use of the form feature provided by OCA. And what started as a convenience function developed into a full-blown user interface for Semantic Containers that also handles the connection to the TDA (used in Semantic Containers and the PDS).

From a technology perspective the DataBud is a single-page Vue.js application that is packaged together with Semantic Containers or can also be run stand-alone with nginx in a dedicated Docker container.

2.3.3 Data Sharing Container

Semantic Container is a platform that provides many features for managing data but it is also easily extensible with domain specific functionality. For the concrete dataflow to demonstrate data sharing additional functionality is needed for researchers to set up an endpoint that manages the complete process of requesting and storing data. This includes the following building blocks:

- maintain participant list,
- send out email to all participants,
- provide information about the requested data, and
- avoiding duplicates from participants submitting their data multiple times.

Relevant features for data sharing that already exist and are used in this dataflow:

- accept incoming data and provide a receipt to the sender,
- manage consent for using the data,
- allow data to be updated later, and
- revoke consent for previously provided data.

2.4 Identity Provider

The DIP solution is aimed to be available globally and thus requires compatibility with different authentication systems. This constraint is contextual and translates into the requirement of allowing multiple identity providers. In this project, we limited our scope to two essential providers:

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

- **eIDAS** stands for **e**lectronic **ID**entification, **A**uthentication and trust **S**ervices. eIDAS is an EU regulation on electronic identification entered into force in 2014. DIP identity providers will have to comply with this regulation to ensure a possible deployment of the DIP solution across EU countries.
- **biometry** to provide access to the system for people with no digital identifiers or who live in places where no electronic identification systems are available. For this purpose we considered the requirements coming from the humanitarian sector through the non-profit organisation iRespond that has a track record in the identification of individuals in low cost environments (e.g. AIDS treatment in Africa, migrant workers in south asia fishing industry). iRespond website[4]

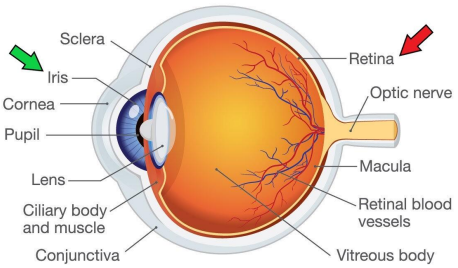
2.4.1 eIDAS

A-Trust[5] is a qualified trust service provider in Austria for electronic certificates and operates on the basis of the eIDAS Regulation. Within the DIP project "Handy-Signatur" (mobile phone signature) is used for individuals to prove their identity. The goals in the DIP project are:

a) DIP delivered through an identity provider subject to eIDAS regulation can be verified internationally.
b) Keep a regulatory watch on the evolution of the european regulations related to digital identity.

2.4.2 Biometry

iRespond is a non-profit technology company active in the humanitarian sector. They provide a digital identity through biometric identification that can be deployed in harsh environments. Their identity solution is based on an identifier generated out of a dual iris scan called the *UNiD*.



Original eye biometrics were based on the retina. But the pattern of vascular tissue and retinal blood vessels found in the back of the eye allows the detection of many health problems and are therefore too invasive for a biometric privacy preserving solution and is one of the rationale behind iRespond's choice of dual iris scan in their solution.

No other information is recorded by iRespond to ensure the absence of personally identifiable information (PII) in their system. Only a hash of the UNiD is available to a verifier. For example, further information can be obtained on their work in Kenya in this Scientific publication co-authored by iRespond "Feasibility and acceptability of an iris biometric system for unique patient identification in routine HIV service in Kenya" in the International Journal of Medical Informatics,

[4] https://www.irespond.org
[5] https://www.a-trust.at/en/

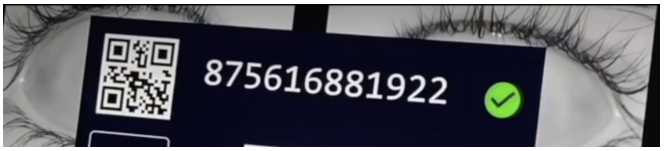
FUNDED BY NGI
This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

2019[6] *(Note: article reflects the technology as of 2018 and not the current improved level)*

In Phase 2 of the NGI DAPSI project we integrate iRespond as a trusted biometric service provider. This extends DIP use case to the humanitarian sector dealing with vulnerable populations in unstable environments.

Biometric identification is, by definition, the most human centric identification system. Thus it offers high hopes in the digital space for safe and privacy preserving solutions. But biometry also introduces high risks particularly when it is used in conjunction with official ids (e.g. passports) or sensitive data (e.g. health status). We adopted a safe approach following the resolution adopted in 2005 International Conference of Data Protection and Privacy Commissioners[7]. Further information about the complexity of the subject can be found in the ICRC Handbook on Data Protection in humanitarian action[8].

In Phase two we have therefore included the iRespond privacy preserving proprietary solution for identification in its simplest form. Leaving further research and developments to a specific project on the topic.



dual iris scan for identification *Generation of a unique identifier*

The unique identifier generated by iRespond (UNiD) is not correlated with any other data. Any subsequent dual iris scan will return the same UNiD. The UNiD is the data element that is being transported in the DIP project.

But to reach its goal of confirming the identity of individuals that have no other means, a system has to ensure that biometric data can only be collected and read with the direct participation of the subject. Related to data portability, two challenges have to be addressed:

1. Biometric template protection. *Can its security be ensured while keeping user control ?*

[6] https://www.sciencedirect.com/science/article/abs/pii/S138650561930855X
[7] available at http://privacyconference2011.org/htmls/adoptedResolutions/2005_Montreux/2005_M4.pdf
[8] see chapter 8 : https://www.icrc.org/en/data-protection-humanitarian-action-handbook

FUNDED BY NGI
This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

2. Biometric sample data use. *How do we prevent correlation and other misuse of biometric data ?*

These challenges have been considered in this project and further work is needed to safely develop an end-to-end protocol. We describe below the current status of the development:

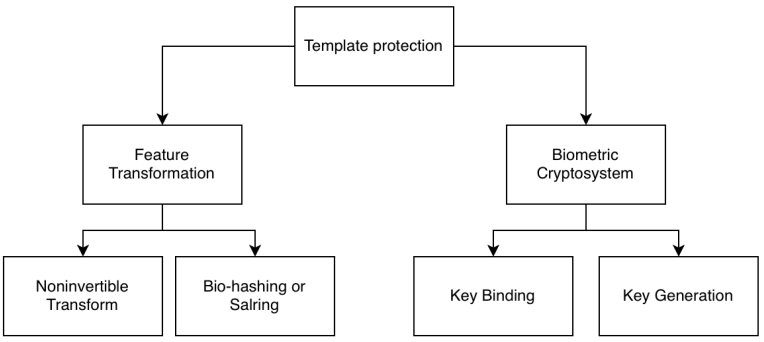
1. **From biometric sample to template:** a biometric sample, the data obtained by the biometric system's capture device, can originate from very different sources. It could be an image of the shape of an individual's hand to his finger, iris, or retina, or a recording of his voice. Alternatively, it could also be behavioral data. This data then becomes a master profile from which the unique features of the individual are extracted, analyzed, and then converted into a mathematical file. This mathematical file comes to be known as the biometric template.
2. **From template to database:** A biometric template is a digital representation of the unique features that have been extracted from a biometric sample and is stored in a biometric database. These templates are then used, through a Biometric Service Provider (BSP) and scanners, in the biometric authentication and identification process. The vulnerabilities of biometric templates are listed below:

    - **Identity theft:** A template can be replaced by an impostor's template.
    - **Impersonation:** The stolen template can be relayed to the matching module to gain unauthorized access to the system. A physical spoof can be created from the original template.
    - **Correlation***:* If not properly secured, biometric templates can be used by adversaries to cross-match across different databases to covertly track a person without their consent.

The human-centric data portability approach of the DIP project aims at:

⇒ keep the user in control of the storage of template instead of the BSP

⇒ through user controlled template *feature transformation*, de-correlate usage of biometric data used in different contexts



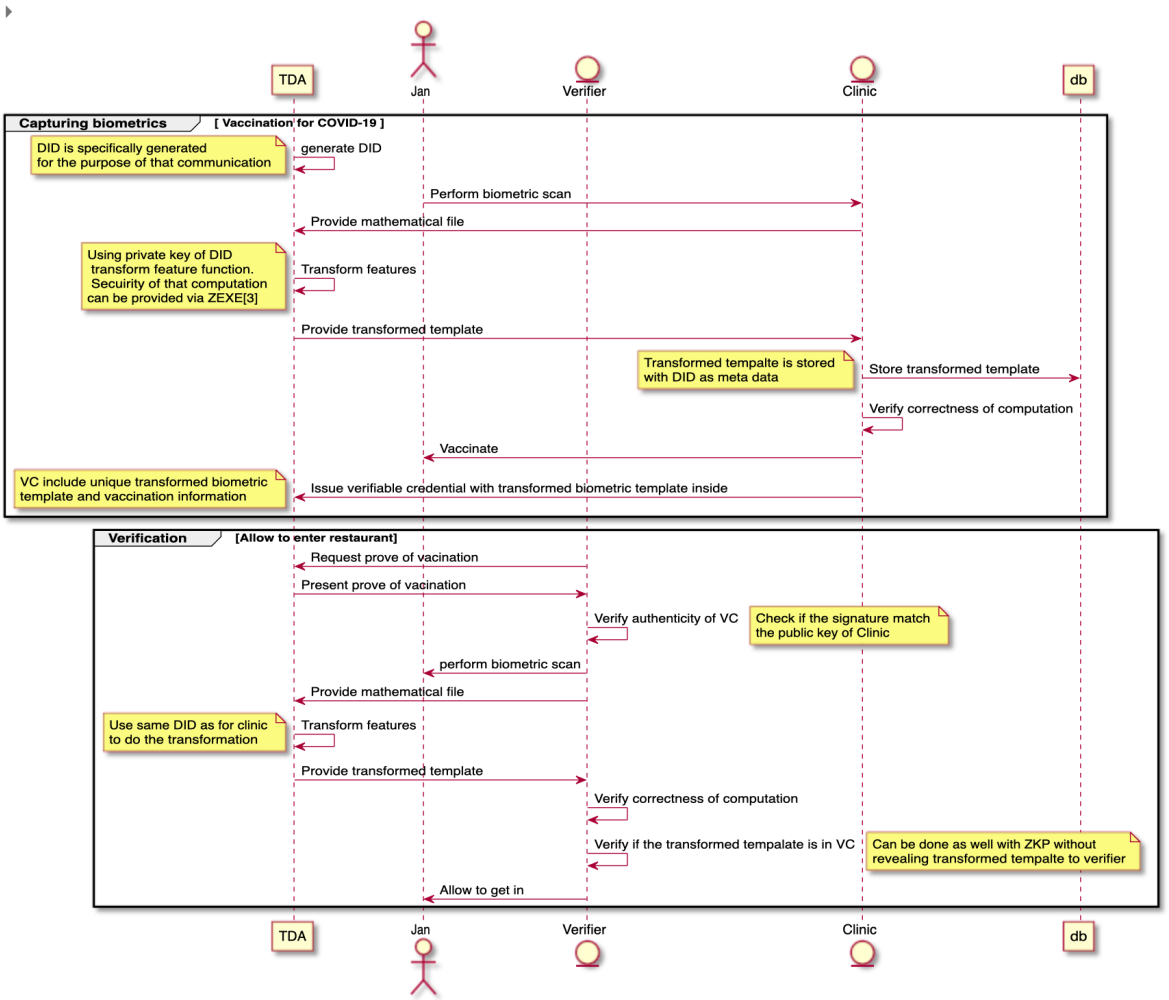
FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

By *Feature Transformation* we mean using a transformation function to alter the features. The parameters of the transformation function are usually derived from a random key or password. Thus opening the possibilities to user control on both the collection and verification of biometric data. The envisaged process is described in the following sequence diagram:

Actors

User (subject):	Jan - Human being, person to be authenticated
TDA:	Jan's Trusted Digital Assistant
Clinic:	Organization that issue the original biometric sample
db:	Biometric template storage
Verifier:	Third party using Jan's biometry for authentication



FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

3 Interfaces & Workflows

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

3.1 Data Structure & API

An important aspect in developing the Digital Immunization Passport is the use of open data structures and APIs to be interoperable with other Personal Data Stores (PDS). This section describes the data exchange format (JSON structure) of records exchanged between the PDS and Semantic Containers as well as API endpoints used for standard operations.

To exchange records within DIP the following types of information are necessary:

- content: this is the actual payload and is represented as attribute–value pairs and array data types
- metadata: a number of attributes are relevant for each record (or group of records); (note: attribute names in `Courier font`)
    - unique ID for a record; in fact multiple IDs may be used to identify a record - in the case of DIP these are:
        - ID: consecutive number of a data record assigned by the database (`id`)
        - DRI (decentralized resource identifier): basically a hash value of the serialized content that unambiguously describes the record (`dri`)
    - Schema: defines the structured description of the content (since this is also provided as a record in a registry it can be itself described by a DRI → attribute name `schema_dri`)
    - Usage Policy: a machine readable description to express the data subjects' consent in using this record[9] (`usage-policy`)
    - Provenance: a machine readable description of the origin and processing steps that lead to the current state of the record using PROV-O[10] (`provenance`)
    - timestamps: when was the record created (`created_at`) and last updated (`updated_at`)
- validation: since content and meta data can contain sensitive or even legal information (e.g., usage policy) it can be necessary to provide a trust anchor, i.e., information that allows to independently verify a certain state of content and metadata at a given time; the following validation methods are available provided by the OwnYourData Notary service[11]:
    - distributed ledger: store the hash value of content and metadata in a blockchain (here: Ethereum mainnet) and provide the address within the ledger as distributed and independent verification source

[9] https://www.specialprivacy.eu/images/documents/SPECIAL_D21_M12_V10.pdf
[10] https://www.w3.org/TR/prov-o/
[11] https://notary.ownyourdata.eu

FUNDED BY NGI
This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

- trusted timestamp: as described in RFC 3161[12] the hash value of content and metadata is signed together with the timestamp from an eIDAS conform trusted timestamping authority

A record is a nested object of content, metadata, and validation as depicted in the Figure below. Based on the parameters in a request one or multiple parts are returned.

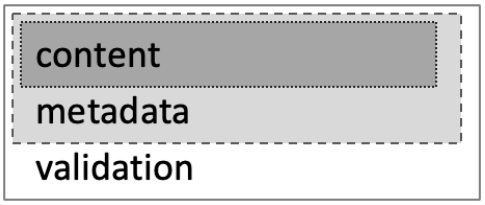


Figure 3.1: Data Structure

To access a data store (PDS or Semantic Container) the following groups of API endpoints are described below:

- Authorization
- Retrieve a Single Record
- Retrieve a List of Records
- Retrieve all Schemas
- Create a Record
- Update a Record
- Delete a Record

Authorization

The OAuth 2.0 client-credential grant flow is used for server-server communication as described in https://auth0.com/docs/flows/client-credentials-flow

Request:

`POST /oauth/token`

Body: provide `client_id`, `cliend_secret`, and optionally `scope`

Example:

```
curl -X POST -s -d grant_type=client_credentials -d scope=admin \
     -d client_id=c196066b21eeb9df20056447467d7132696d7558a3208610e0dab6941a9434b8 \
     -d client_secret=fb6f99f75e2d37943c4a8f9196dd07ed96daeef525a88d03c2246093074973c6 \
     https://dip-clinician.data-container.net/oauth/token
```

Response:

```
{
  "access_token":"ef2438ba6c0b833ac1d5007e565ab5c5dcc1afe6038b586082548585786af215",
  "token_type":"Bearer",
```

[12] https://www.ietf.org/rfc/rfc3161.txt

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

```
  "expires_in":7200,
  "scope":"admin",
  "created_at":1604930855
}
```

Note regarding CEPS compatibility: section "Authentication" also specifies OAuth 2.0 but uses the password grant-flow

Retrieve a Single Record

To retrieve information of a single record two options are available to select this record (either through ID or DRI) and a number of options are available which information is provided: content, metadata, and/or validation.

Request:

```
GET /api/data/:ref?p=[id|dri]&f=[plain|meta|full|validation]
```

`:ref` is the reference to identify a record

URL parameter `p` specifies if the reference is an ID or DRI

URL parameter `f` specifies the expected return type:

- plain: only the content is returned
- meta: only the metadata of the record is returned
- full: content and metadata is returned (default)
- validation: content, metadata, and validation information is returned

Example:

```
curl -H "Authorization: Bearer $TOKEN" \
     "https://dip-clinician.data-container.net/api/data/1?p=id&f=plain"
```

Response:

```
{"first_name":"Christoph","last_name":"Fabianek","gender":"male"}
```

Note regarding CEPS compatibility: section "Reading data" specifies the endpoint `GET /ceps/read/{table-identifier}/{record-id}` but does not yet support querying by multiple identifiers and providing options for different return types.

Retrieve a List of Records

To retrieve multiple records three options are available.

Request:

- query by schema_dri
  `GET /api/data?schema_dri=:schema_dri&f=[plain|meta|full|validation]&page=1`

- query by table (not available in Semantic Containers)
  `GET /api/data?table=:table_name&f=[plain|meta|full|validation]&page=1`

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

- query all records
  `GET /api/data?f=[plain|meta|full|validation]&page=1`

Pagination: since the response can include a large number of records the returned array is restricted to the first 20 entries; use the URL parameter page to retrieve subsequent entries; in the Header response the following additional information is provided Link (as described in RFC-8288[13]), Current-Page, Page-Items, Total-Pages, and Total-Count.

Example:

```
curl -H "Authorization: Bearer $TOKEN" \
     "https://dip-clinician.data-container.net/api/data?f=plain"
```

Response:

```
[{"first_name":"Christoph","last_name":"Fabianek","gender":"male"}]
```

Note regarding CEPS compatibility: section "Querying data" specifies the endpoint `GET /ceps/query/{table-identifier}[?{attribute}={value}[&attrib=val]..]`

Retrieve all Schemas

To retrieve all available schemas (i.e., a list of unique schema_dri) in a data store the following endpoint is available.

Request:

`GET /api/meta/schemas`

Example:

```
curl -H "Authorization: Bearer $TOKEN" \
     "https://dip-clinician.data-container.net/api/meta/schemas"
```

Response:

```
["gffA2i9tCexTwQ1S6JsXxJ8JEMHfTdaMtggBjX6jvF8N"]
```

Note regarding CEPS compatibility: this functionality is relevant for compatibility with OCA and currently not included in CEPS

Create a Record

To insert a new record in a data store the following endpoint is available.

Request:

`POST /api/data`

[13] https://tools.ietf.org/html/rfc8288

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

Body:

```
{
  "content":{"attribute": "value", … },
  "schema_dri":"123…",
  "dri":"345…",
  "mime_type":"application/json",
  "table_name":"oyd.dip"
}
```

Note that the actual payload attributes are wrapped in the content attribute and metadata (`schema_dri`, `dri`, `mime_type`, `table_name`) are outside; all metadata attributes are optional. If `table_name` is omitted the OwnYourData Data Vault uses "default".

Example:

```
curl -H 'Content-Type: application/json' -H "Authorization: Bearer $TOKEN" \
     -d '[{"content":{"first_name":"x","last_name":"y","gender":"1"},
           "schema_dri":"gffA2i9tCexTwQ1S6JsXxJ8JEMHfTdaMtggBjX6jvF8N"}]' \
     -X POST "https://dip-clinician.data-container.net/api/data"
```

Note that providing an array of objects creates multiple records.

Response:

```
{
    "receipt":"b7ca743ec49b831feb6b516cf53621cb82aa985129c99ffb82780db5922d3d5d",
    "serviceEndpoint":"https://dip-clinician.data-container.net",
    "revocationKey":"6bbebc45e6b0b03d9d0779ff8fc7d974"
}
```

Note that the `receipt` attribute provides a unique token for requesting information (GDPR Art. 15) about the sent data (using the `GET /api/receipt/:receipt`), `serviceEndpoint` is the URL to send any subsequent queries to, and `revocationKey` is the necessary token if provided data should be deleted (GDPR Art. 17).

Note regarding CEPS compatibility: section "Writing data" specifies the endpoint `POST /ceps/write/{table-identifier}`

Update a Record

To update an existing record in a data store the following method is available.

Request:

`PUT /api/data/:ref?p=[id|dri]`

Body: same attributes and structure as in "Create a Record"

Response: same as in the section above "Create a Record"

Note regarding CEPS compatibility: section "Updating data" specifies the endpoint `PUT /ceps/update/{table-identifier}/{record-id}`

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

Delete a Record

To remove an existing record in a data store the following method is available.

Request:

```
DELETE /api/data/:ref?p=[id|dri]
```

Example:

```
curl -H "Authorization: Bearer $TOKEN" \
     -X DELETE "https://dip-clinician.data-container.net/api/data/1?p=id"
```

Response:

```
{"id": 123, "dri": "abc…"}
```

Note: the dri attribute is only shown when it was available in the deleted record

Note regarding CEPS compatibility: section "Data deletion" specifies the endpoint `DELETE /ceps/update/{table-identifier}/{record-id}`

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 lab side to provide such a credential.

3.2.1 Preparation

This section describes the necessary steps to set up the infrastructure for user and clinician and Figure 4.1 depicts the sequence diagram.

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

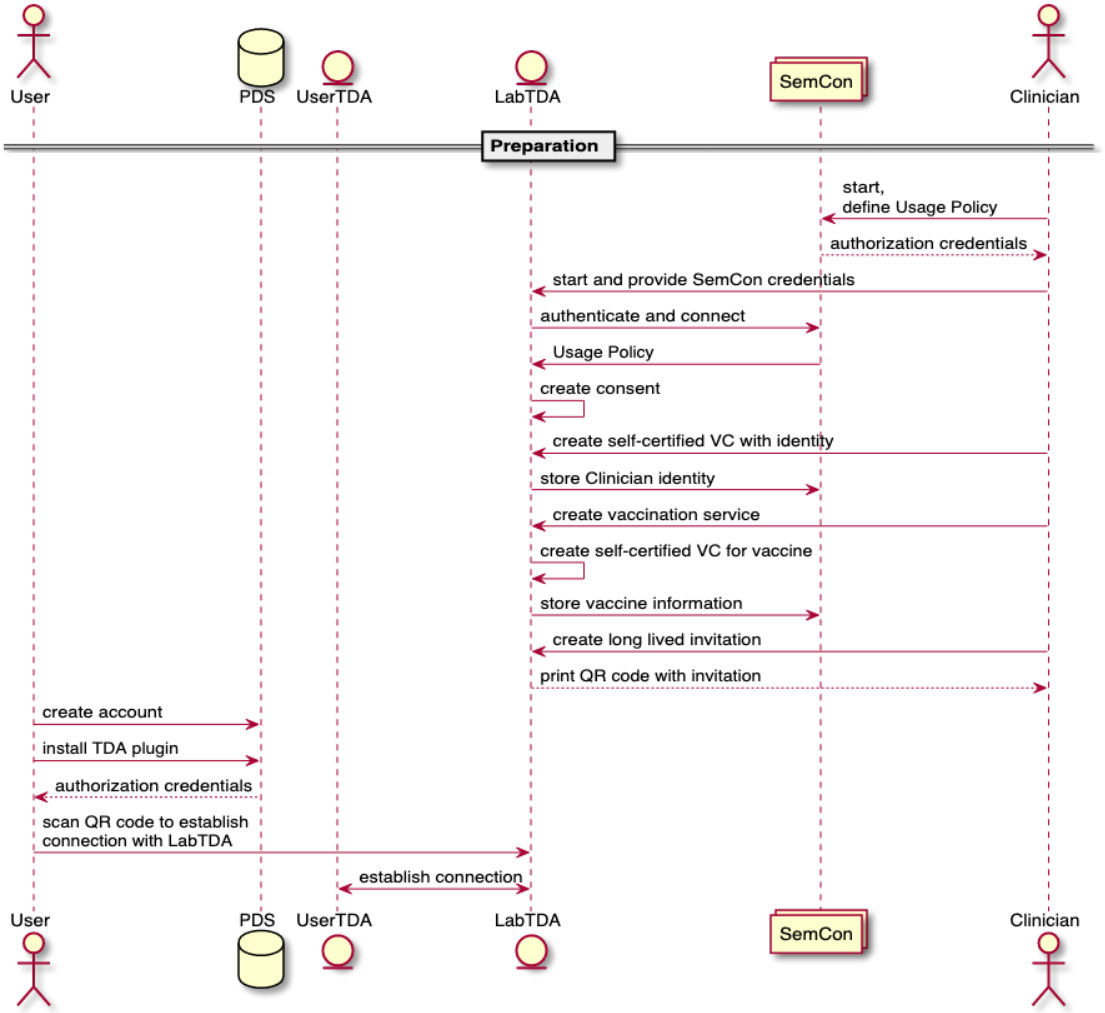


Figure 3.2: Prepare creating verifiable credential

3.2.2 Gather Information

This section describes the process of the user to gather information about clinician and vaccination and prepares the information to submit a vaccination request.

FUNDED BY NGI
This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

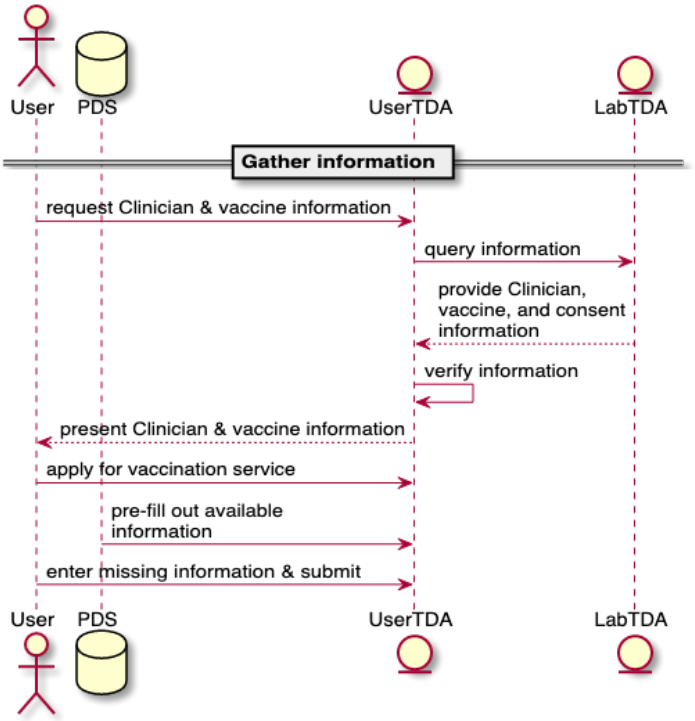


Figure 3.3: Gather information to request vaccination

3.2.3 Create Verifiable Credential for Vaccination

This section describes the process of the clinician to verify the user identity, performing the vaccination, and issuing a verifiable credential.

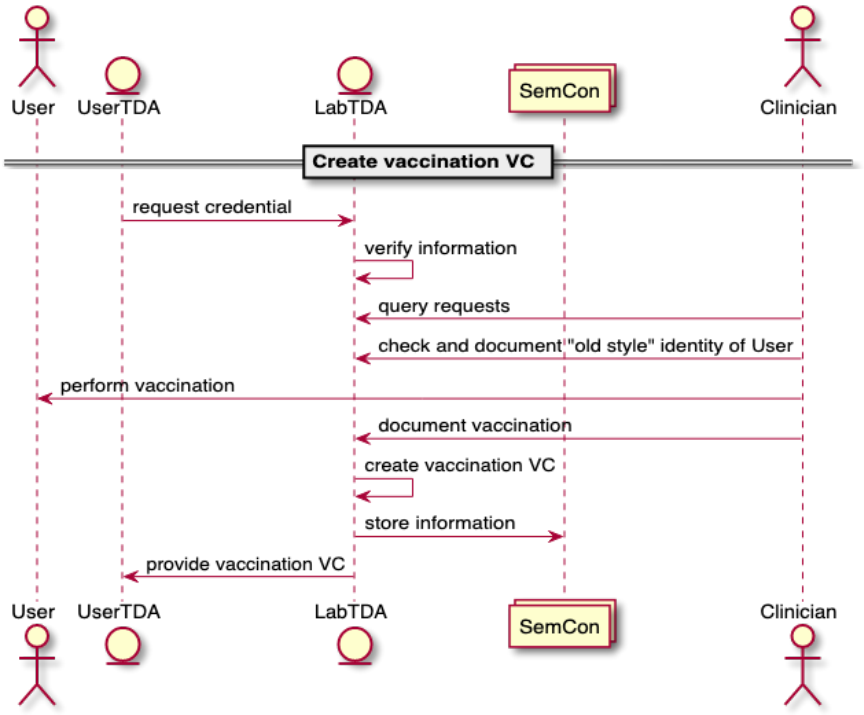


Figure 3.4: Create Verifiable Credential for Vaccination

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

3.2.4 Human-centric Verifiable Credential Management

This section describes a user that manages verifiable credentials in a Personal Data Store.

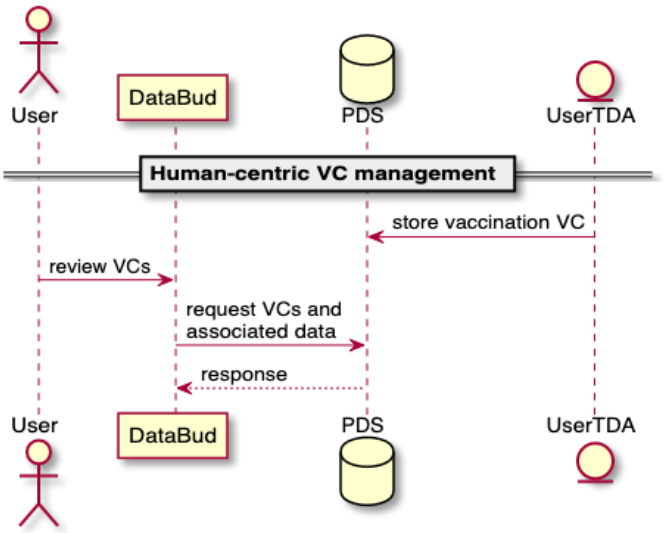


Figure 3.5: Human-centric Verifiable Credential Management

3.3 Proofing Immunization Status

3.3.1 Gather Information

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

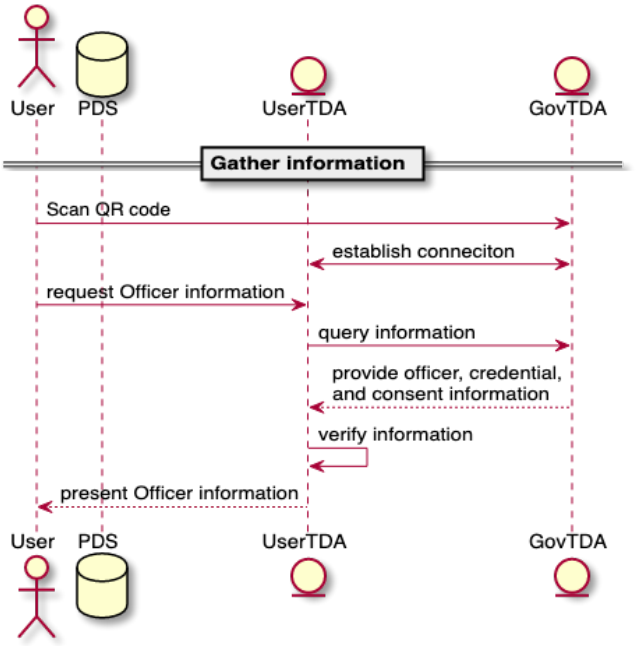


Figure: 3.6: Gather Information

FUNDED BY NGI
This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

3.3.2 Present Credentials

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

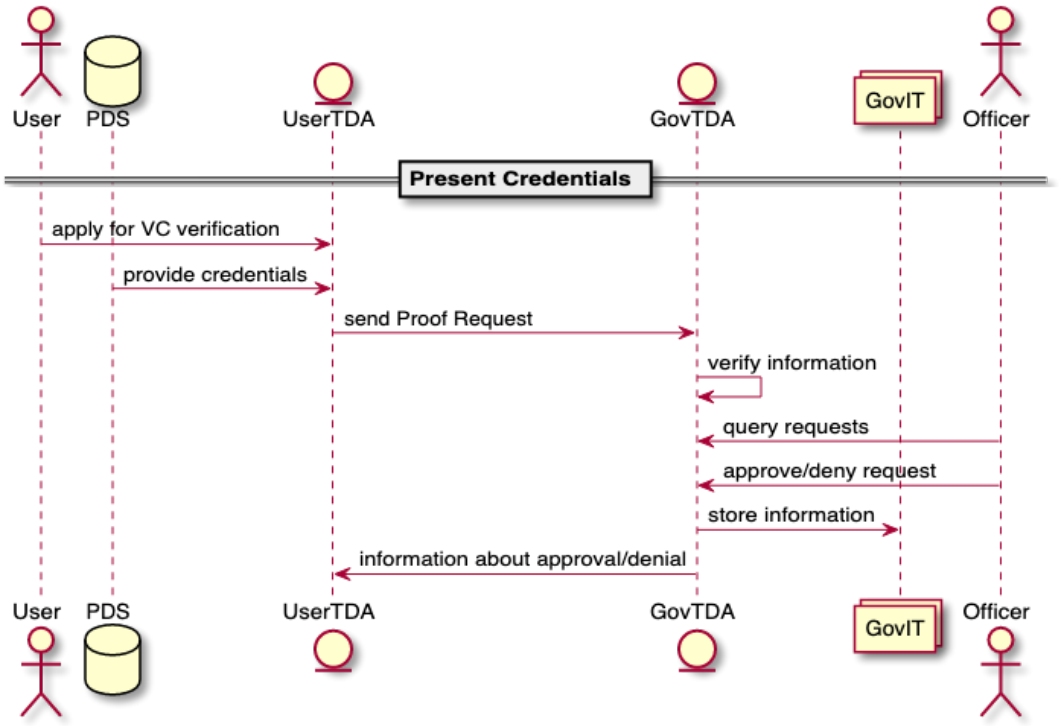


Figure: 3.7: Present Credentials

3.3.3 Review Information

This section describes a user that manages presented credentials and responses in a Personal Data Store.

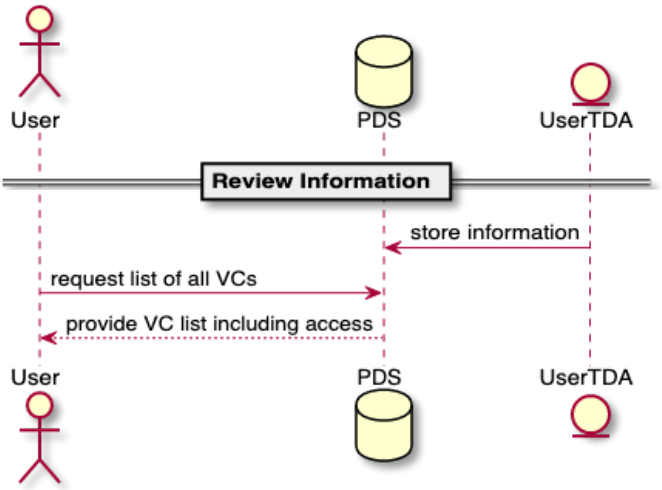


Figure: 3.8: Review Information

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

3.4 Sharing Data

3.4.1 Contact Participants

This section describes a research inviting users to share data.

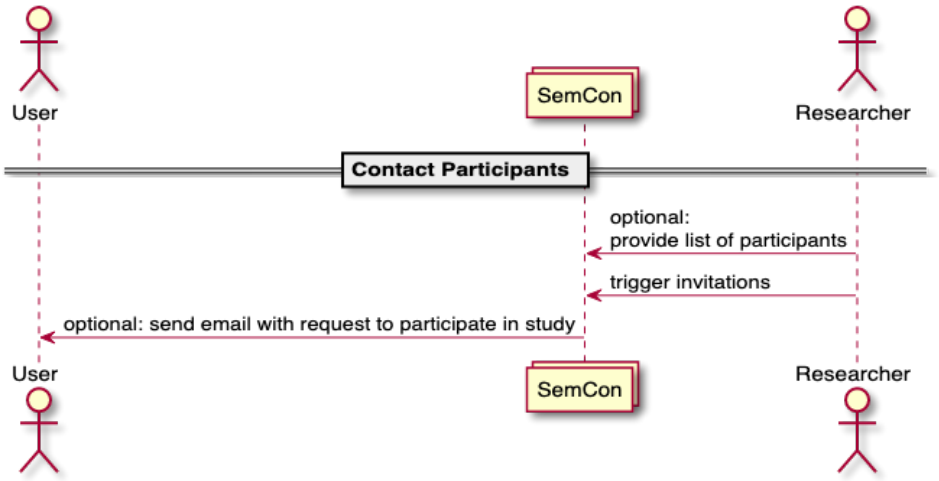


Figure: 3.9: Contact Participants

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.

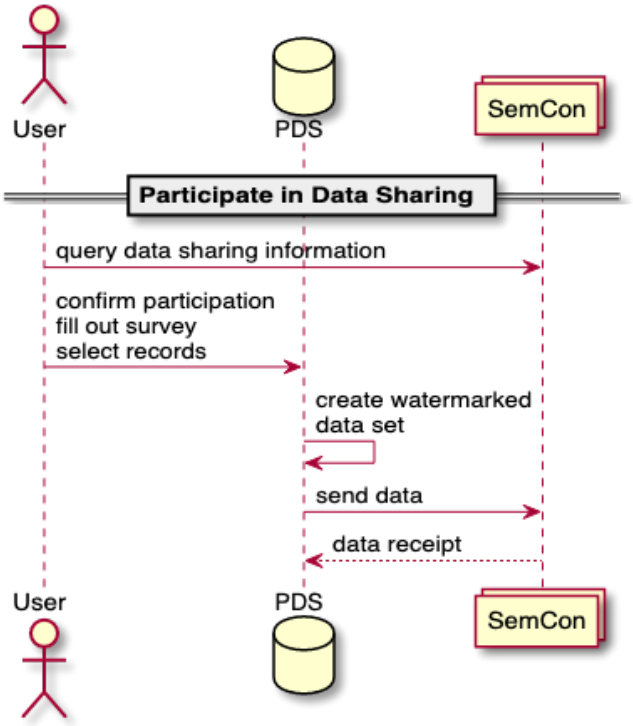


Figure: 3.10: Participate in Data Sharing

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

3.4.3 Tracing Data

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

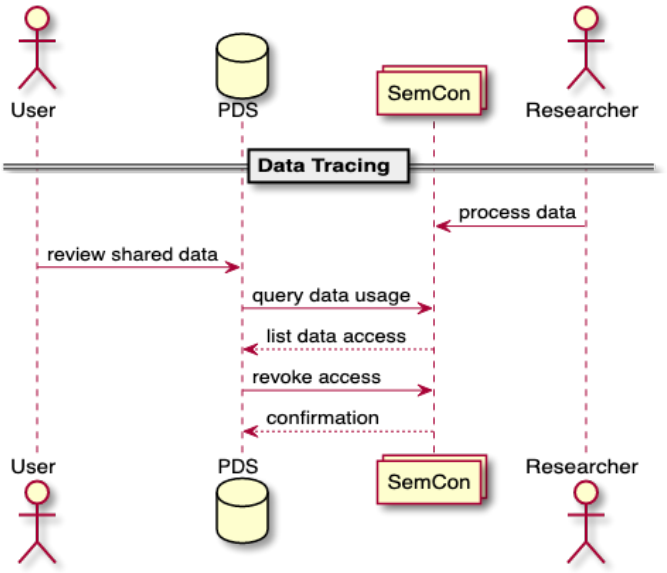


Figure 3.11: Tracing Data

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

4 System Testing

4.1 Test Cases Phase #1

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

1) Creating a Verifiable Credential

Prerequisites:

- OwnYourData App installed on smartphone
- Semantic Container & TDA bundle for Clinician deployed

Tests:

a) Setup
<u>Individual</u>
- OYD Data Vault Account and Plugins installed: TDA, DataBud
- identity of individual set

<u>Clinician</u>
- Semantic Container & TDA started
- identity of Clinician set
- vaccination info set

b) Individual requests credential
- Individual scans QR from Semantic Container
- show information about Clinician and Vaccination
- trigger Vaccination Request and provide consent for managing personal information
- provide missing information (i.e., personal information)

c) Clinician performs treatment
- show list of treatment requests and select item
- document identification detection (iRespond can be used here) and confirm treatment

d) Store Verifiable Credential
- create Verifiable Credential (automatically triggered after treatment documentation)
- transfer data to OYD Data Vault
- Show Verifiable Credential with DataBud

Verification:

- Treatment is documented in Semantic Container
- Verifiable Credential is listed in Personal Data section

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

2) Proofing Immunization Status

Prerequisites:

- OwnYourData App installed on smartphone
- Semantic Container & TDA bundle for Officer deployed
- Verifiable Credential stored in OYD Data Vault

Tests:

e) Setup
<u>Individual</u>
- OYD Data Vault Account and Plugins installed: TDA, DataBud
- identity of individual set
- Verifiable Credential for Yellow Fever available

<u>Officer</u>
- Semantic Container & TDA started
- identity of Officer set

f) Individual presents Verifiable Credential
- Individual scans QR from Semantic Container of Officer
- show information about Officer including consent information
- trigger providing Verifiable Credential

g) Officiers verifies credential
- show list of provided credentials
- document identification detection and confirmation/denial
- confirm / deny approval

h) Store confirmation/denial
- store result of Officer response in OYD Data Vault
- show confirmation / denial linked to Verifiable Credential in OCA Viewer

Verification:

- Verification is documented in Semantic Container
- Officer and verification information is stored in OYD Data Vault

3) Sharing Data with Pharmaceutical Industry

Prerequisites:

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

Tests:

i) Setup
<u>Individual</u>

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

- OYD Data Vault Account and Plugins installed: Data Sharing & Consent

Organization

- Semantic Container to receive individually shared data
- define Usage Policy for received data
- list of contacts participating in data sharing

j) Invite to data sharing
- send email to planned participants with URL of Semantic Container for data sharing
- retrieve and show information from Semantic Container in OYD Data Vault
- select data to be shared

k) Send data
- apply digital watermarking to selected data
- send watermarked data together with Usage Policy to Semantic Container
- store information about shard data

l) Manage shared data
- retrieve information about shared data
- revoke consent for shared data
- identify if a dataset is own dataset (based on digital watermark) and whom it was shared with

Verification:

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

4.2 End-to-End Tests

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 of DIP. The following graphic outlines the test system setup.

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

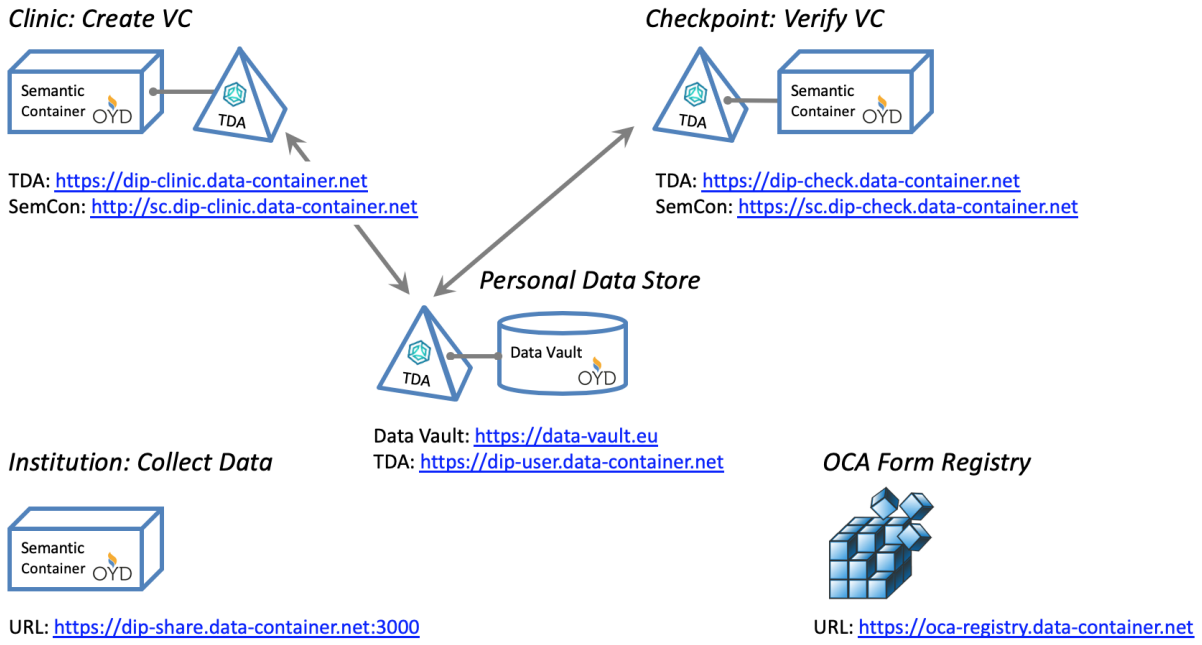


Figure 4.1: DIP Test System Setup

4.2.1 Test Results

This section describes performed tests and results in the first phase of the DIP project.

1) Creating a Verifiable Credential
   Prerequisites:
   - Semantic Container / DataBud: v0.10.15
   - TDA:
     - new UI missing
   - Data Vault: n.a.

   a) Setup
      - User: Jan Kowalsky
        - login Data Vault
        - open TDA
      - Clinician
        - login DataBud
        - open TDA
        - create consent - re-use Usage Policy from Data Store
        - setup vaccination information (service)
   b) User: scan Clinician QR
   c) User: show information about clinician and vaccination
   d) User: trigger Vaccination request
      OCA Schema: ImmunizationPassport
   e) Clinician: see pending application
   f) Clinician: document identification not included

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

2) Verifying a Verifiable Credential

a) Setup
   - User: Jan Kowalsky
     - login Data Vault
     - open TDA
   - Officer
     - login DataBud
     - open TDA

b) User: scan Officer QR to establish connect
c) Officer: click Ask for Permission and specify OCA Schema
d) Officer: verifies received information - not available yet
e) Officer: confirms/denies request - not available yet

4.3 Test Cases Phase #2

To demonstrate additional functionality of phase #2 requirements in the demo system, the following end-to-end test cases are used for verification.

1) Creating a Verifiable Credential

Prerequisites:

- Semantic Container & TDA bundle for Clinic deployed

Tests:

a) Setup

<u>User</u>

- OYD Data Vault account and plugins installed: DataBud
- identity of user available in OCA Form

<u>Clinic</u>

- Semantic Container & TDA started

b) Configure Clinic Organization and Doctors
- enter Clinic information including LEI (Legal Entity Identifier)
- create 2 doctors linked to the Clinic information
- setup Yellow Fewer and TBE services in TDA
- print QR Code from Clinic

c) User requests credential
- User scans QR from Clinic in User TDA
- select service (TBE) and consent to Usage Policy
- fill out TBE form for initial vaccination (personal information pre-filled)
- trigger Vaccination Request

d) Doctor performs treatment
- show list of treatment requests in Clinic TDA and select item

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

- document identification detection from iRespond in Vaccination Request Form

e) perform and document treatment

f) Store Verifiable Credential
- create Verifiable Credential (automatically triggered after treatment documentation)
- store Verifiable Credential linked to Doctors HR record
- transfer and store data to OYD Data Vault
- Show Verifiable Credential with DataBud

Verification:

- Treatment is documented in Semantic Container
- Verifiable Credential is listed in Personal Data section

2) Proofing Immunization Status

Prerequisites:

- Semantic Container & TDA bundle for Checkpoint deployed

Tests:

a) Setup

<u>User</u>
- OYD Data Vault account and plugins installed: TDA, DataBud
- identity of user available in OCA Form
- Verifiable Credential for Yellow Fever available

<u>Checkpoint</u>
- Semantic Container & TDA started
- identity of Officer set

b) Configure Checkpoint Organization and Officers
- enter Checkpoint information including LEI
- create 2 officers linked to the checkpoint information
- print QR Code from Checkpoint

c) User presents Verifiable Credential
- User scans QR from checkpoint in User TDA
- show information about Officer including consent information
- trigger providing Verifiable Credential

d) Officiers verifies credential
- show list of provided credentials
- document identification detection and confirmation/denial
- confirm / deny approval

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

- link credential for confirmation/denial to Officer record

e) Store confirmation/denial
   - store result of Officer response in OYD Data Vault
   - show confirmation / denial linked to Verifiable Credential in OCA Viewer

Verification:

- Verification is documented and linked to Officer in Semantic Container
- Officer and verification information is stored in OYD Data Vault

3) Sharing Data with Pharmaceutical Industry

Prerequisites:

- Semantic Container from Institution for data sharing setup

Tests:

a) Setup

<u>User</u>

- OYD Data Vault account and plugins installed: Data Sharing & Consent
- personal time series data stored in OYD Data Vault

<u>Institution</u>

- Semantic Container to receive shared data
- defined Usage Policy for received data
- list of contacts participating in data sharing

b) Invite to data sharing
   - send email to planned participants with URL of Semantic Container for data sharing
   - retrieve and show information from Semantic Container in OYD Data Vault
   - select data to be shared

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

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

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

Verification:

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

4) Load Tests

Prerequisites:

- empty Semantic Container at "Clinic" and Shell scripts to simulate load
- Data Vault with test account setup
- TDA setup

Tests:

a) Write data to Semantic Container
   - plot time to write batches of 100 records for small (40 bytes), medium (300 byes), and large (>1.5 kilobytes) objects and iterate 1.000 times

b) Read data from Semantic Container
   - plot time to read single records based on ID or DRI
   - plot time to read group of records based on Schema Base DRI

c) Write data to Data Vault (personal data store)
   - measure time to write data (OCA Form data and Consent data)

d) Read data from Data Vault (personal data store)
   - measure time to read arbitrary records
   - measure time to read relations between arbitrary records

4.3.1 Test Results

Status of system in DIP Phase #2 tests:

1) Creating a Verifiable Credential
Prerequisites: installed and available with the following versions:
   - Semantic Container / DataBud: v0.10.33
   - TDA: v0.5.4 ext-26
   - Data Vault: v0.7.1

a) Setup
   - User: Jan Kowalsky
     i) login Data Vault
     ii) open TDA
   - Clinic
     i) open TDA
     ii) create consent

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

iii) create LEI for clinic
iv) create 2 doctors linked to clinic
v) create "Yellow Fever Vaccination Record" service

b) User requests vaccination credential
- User setup connection
  i) scans Clinic QR Code (create connection between User and Clinic TDA)
  ii) "Yellow Fever Vaccination Record" is available in Service section

2) Load tests
the script for performing this load tests is available on Github here: https://github.com/sem-con/loadtests

a) Write data to Semantic Container
- measure time for writing 100 records of size 40 bytes (small) and repeat 1.000 times - see Figure 4.2

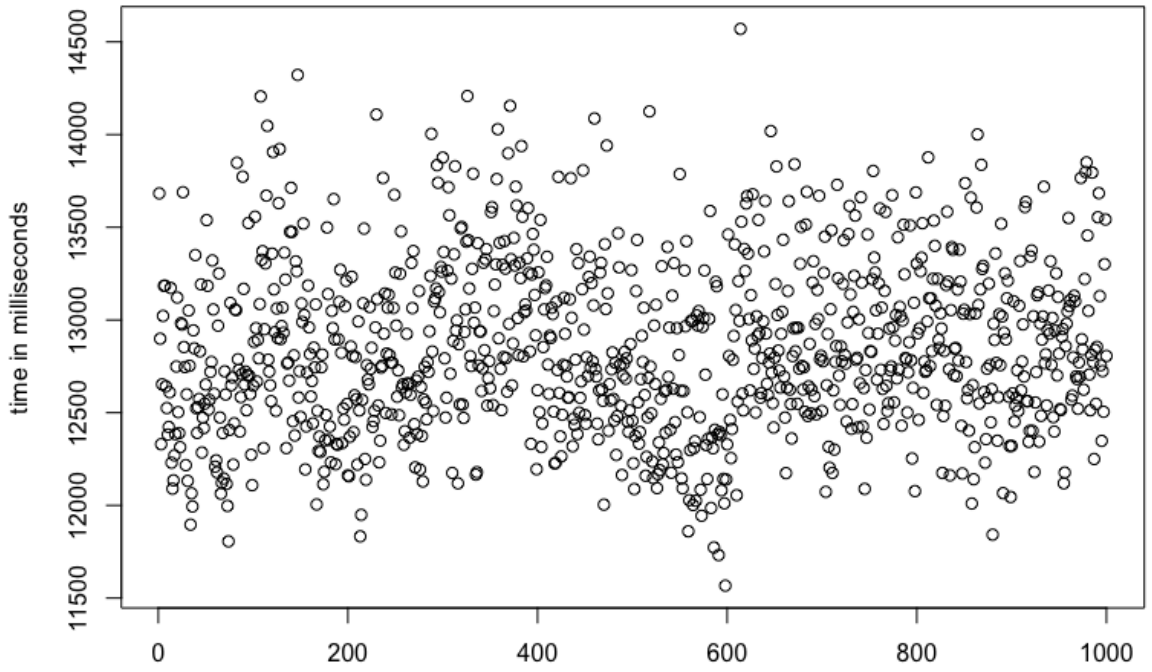


Figure 4.2: time for writing 100 records of small objects (40 bytes)

result: time duration is evenly spread for write operations (mean=12.847ms, std. deviation=463) and does not increase

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

over time with increasing number of records maintained in the container

- measure time for writing 100 records of size 40 bytes (small) and repeat 1.000 times - see Figure 4.3

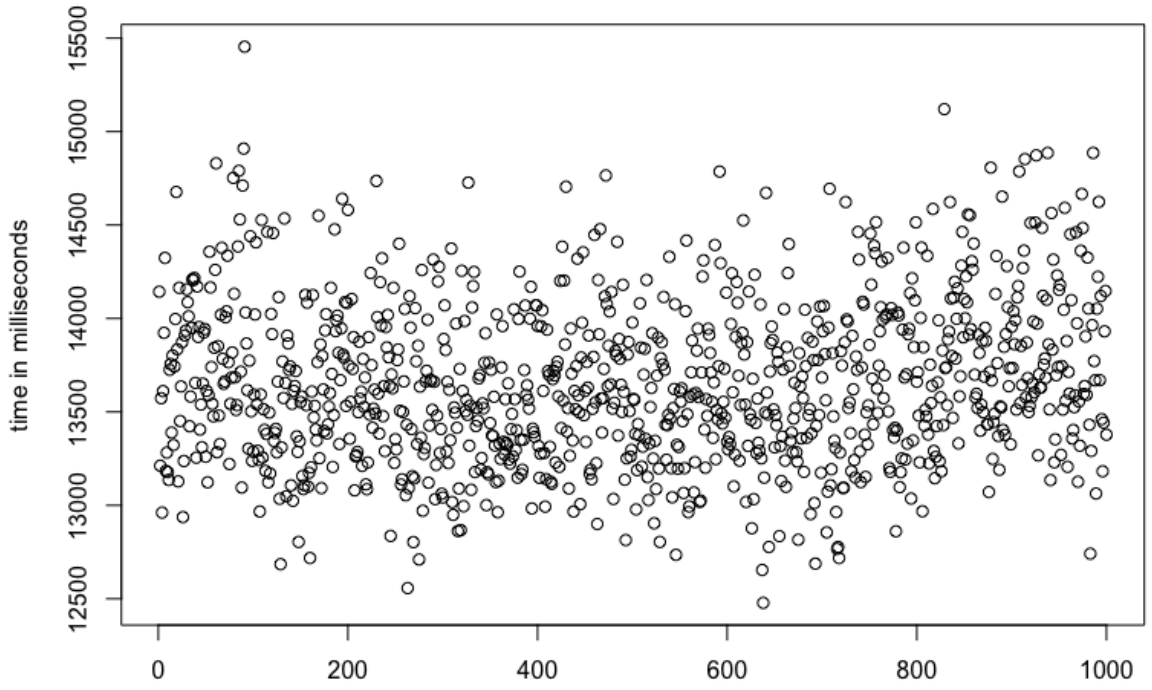


Figure 4.3: time for writing 100 records of medium objects (300 bytes)

result: time duration is evenly spread for write operations (mean=13.656ms, std. deviation=439) and does not increase over time with increasing number of records maintained in the container

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

- measure time for writing 100 records of size 1.5kB (large) and repeat 1.000 times - see Figure 4.4

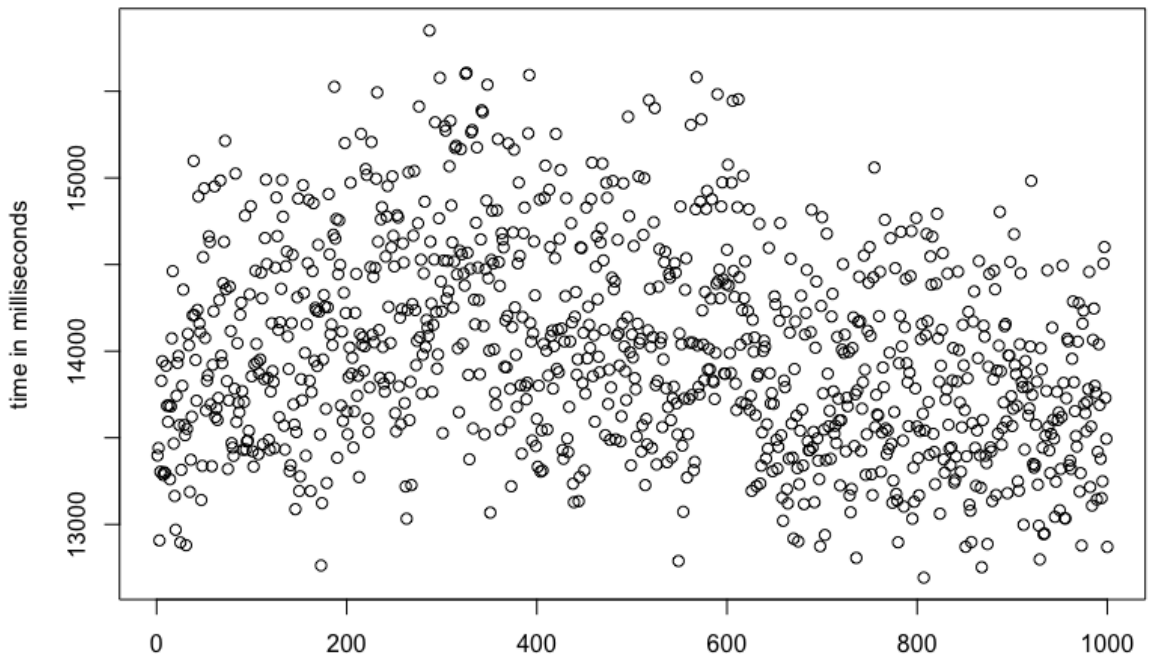


Figure 4.4: time for writing 100 records of large objects (1.5 kB)

result: time duration is evenly spread for write operations (mean=13.993ms, std. deviation=585), similar to previous tests with smaller objects, and does not increase over time with increasing number of records maintained in the container

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

b) Read data from Semantic Container
   - in parallel to writing into the container, tests were performed to read 10 random records after every 100 records - see Figure 4.5

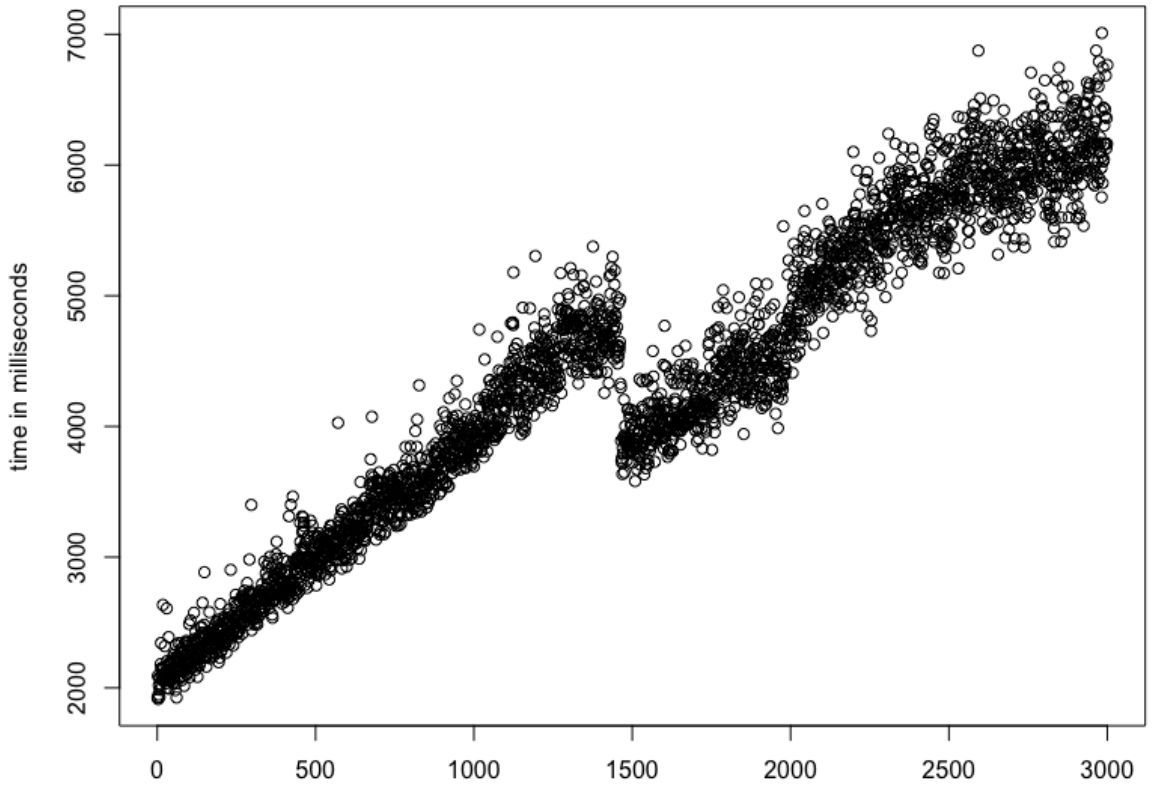


Figure 4.5: time to read 10 random records with a growing number of data stored

result: time duration is increasing almost linearly and takes in the end about 600ms to read a single record in a container with 300.000 records; note the non-linear step at about 140.000 records where another search algorithms from the underlying PostgreSQL takes over

Benchmarking TDA (Trusted Digital Assistant):

Test environment:

- 8GB ram
- Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz

1. Test case – Verifiable Credential issuance

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

Benchmark: run 100 times the Verifiable Credential issuance process between two Agents.

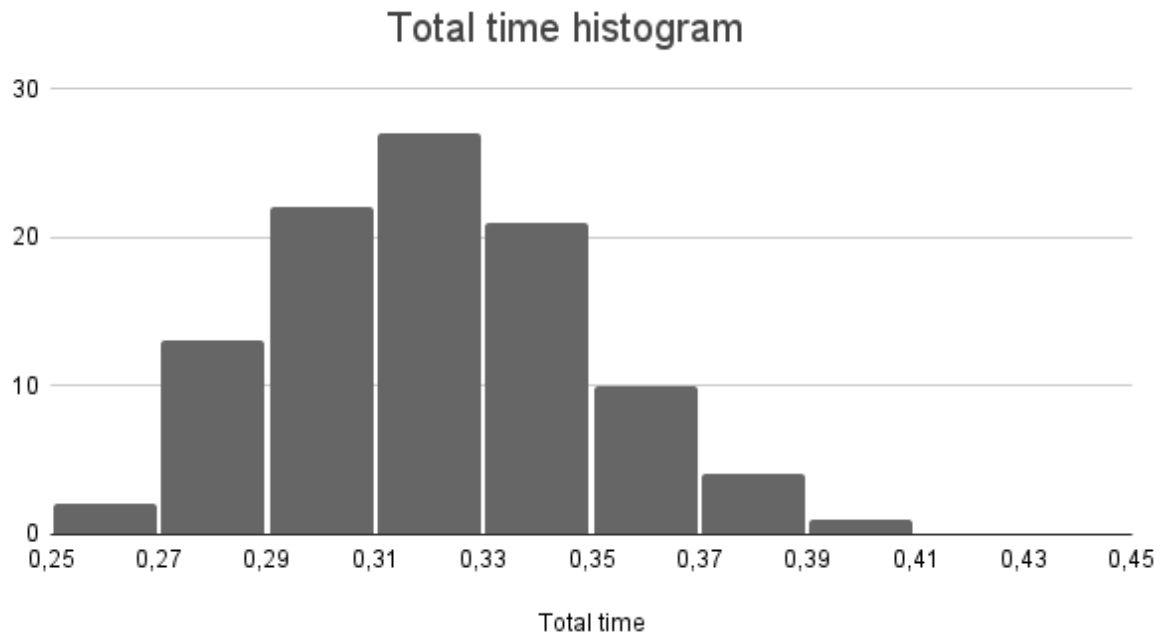


Figure 4.6: The histogram of total time required to complete the process by issuing Agent.

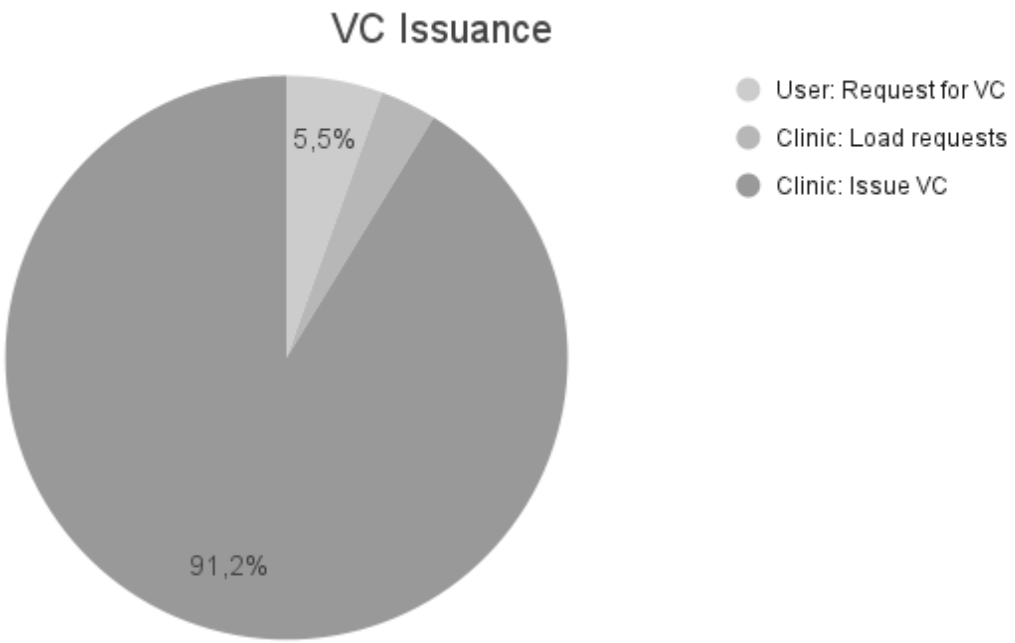


Figure 4.7: The distribution of time spent in each step of the process by both Agents.

2. Test case – Verifiable Credential presentation

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

Benchmark: run 100 times the Verifiable Credential Proofing Immunization process between two Agents.

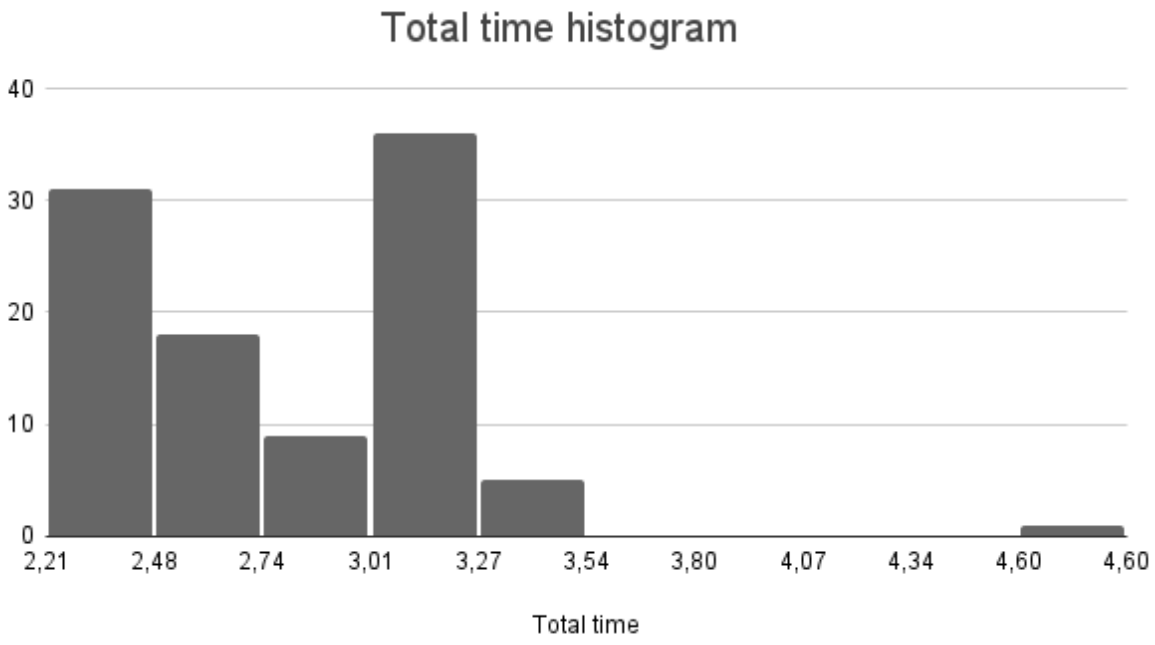


Figure 4.8: The histogram of total time required to complete the process by issuing Agent.

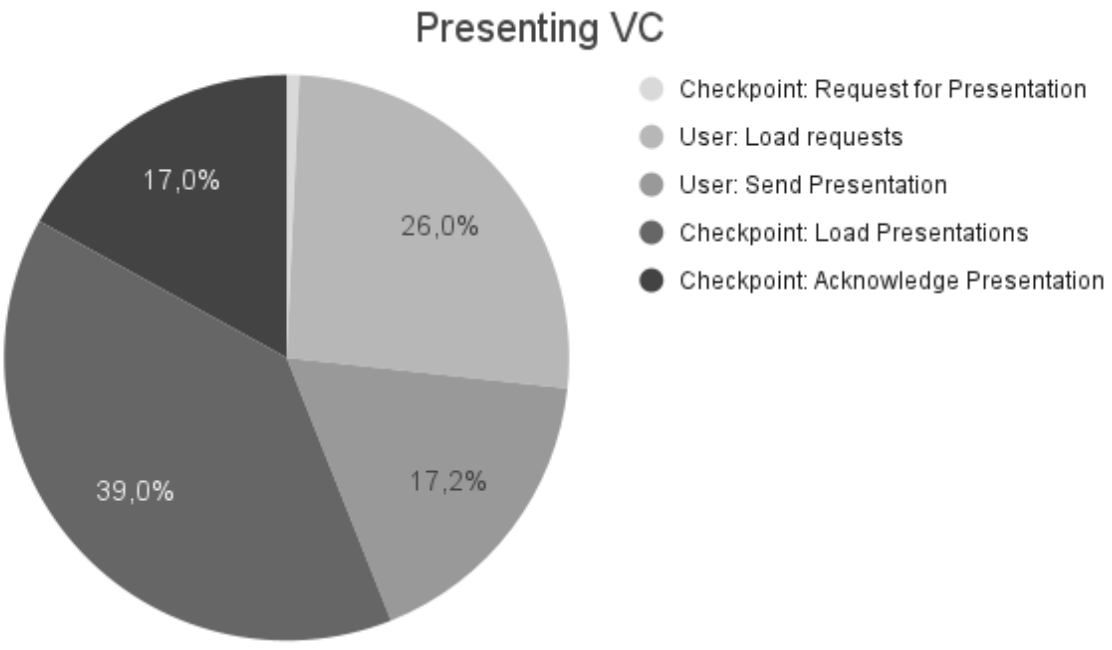


Figure 4.9: The distribution of time spent in each step of the process by both Agents.

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498

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
Data Vault (PDS)	https://github.com/OwnYourData/oyd-pia2
Semantic Container	https://github.com/sem-con/sc-base
DataBud	https://github.com/OwnYourData/oyd-databud
TDA	https://github.com/THCLab/aries-cloudagent-python https://github.com/THCLab/aries-services-plugin https://github.com/THCLab/tda-web-client
OCA (Forms)	https://github.com/THCLab/oca-form https://github.com/THCLab/oca-search-engine

5.2 Outlook

In this project we demonstrate the secure and private exchange of credentials between organizations and individuals as well as individuals sharing data with organizations. With technologies like decentralized identifiers and Personal Data Stores a new way of managing personal data emerges and we are just at the beginning of vast opportunities in this space.

OwnYourData and the Human Colossus Foundation are already in concrete talks with commercial and non-commercial organizations as well as government bodies to put this work into real-life scenarios, and NGI DAPSI provided us with a great opportunity to kick-start this process.

FUNDED BY NGI

This project has received funding from the European Union's H2020 research and innovation programme under Grant Agreement no 871498