Persistent Personal Data Vaults Empowering a Secure and Privacy Preserving Data Storage, Analysis, Sharing and Monetisation Platform

D3.1
Security, Privacy and Trust Bundles - Version 1

Editor(s)	Alexander Köberl
Lead Beneficiary	Infineon Technologies Austria AG (IFAT)
Status	Final
Version	1.0
Due Date	30/06/2021
Delivery Date	12/07/2021
Dissemination Level	PU

This deliverable has been submitted to the EC and is pending approval

DataVaults is a project co-funded by the European Commission under the Horizon 2020 Programme (H2020-ICT-2019-2) under Grant Agreement No. 871755 and is contributing to the BDV-PPP of the European Commission.

Project	DataVaults – 871755
Work Package	WP3 – Bundles for Secure Data Sharing and Access, Privacy and Trust Preservation and IPRs Management
Deliverable	D3.1 – Security, Privacy and Trust Bundles - Version 1
Editor(s)	IFAT – Alexander Köberl
Contributor(s)	ASSENTIAN - Ilesh Dattani ATOS - Miguel Angel Mateo Suite5 - Sotiris Koussouris Tecnalia - María José López Ubitech - Athanasios Giannetsos UNISYSTEMS – Georgios Georgakakos
Reviewer(s)	ETA – Marina Cugurra UBITECH - Giannis Ledakis

Abstract	This deliverable represents an auxiliary document, describing the progress of the code and component development under WP3.
Disclaimer	The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Communities. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein. © Copyright in this document remains vested with the DataVaults Partners

Executive Summary

This deliverable represents an auxiliary document to summarize the progress of code development in the individual components for both the Personal App and Cloud Platform of DataVaults. It collects the direct outcome of all WP3 tasks in a deliverable of type OTHER, which will be updated in future versions. The intermediate results reflect the development progress made in months 9-18 within this work package.

The components that are presented in this deliverable are split into DataVaults Personal App and Cloud Platform. This is done to group related components and clearly assign the responsibility to the target platform.

Personal App:

- **Trusted Platform Module (TPM) Interface** to provide a hardware anchor for privacy and attestation services.
- **Sharing configurator** to provide a user interface to collect the data and apply the selected sharing operations.
- **Data Request Service Resolver** for receiving asynchronous inquiries for additional data and updated sharing preferences.
- **Personal Wallet** for managing the compensation in a privacy-preserving manner.
- **Blockchain Security 2Go Starter Kit** as interface to secure key management and transaction signing with smart cards.
- **Data Anonymizer** for obfuscating the data in order to preserve the privacy of the user.
- **Attribute Based Encryption Engine** to ensure the confidentiality of the data.
- **Access Policies Editor** for configuring conditions for data sharing.

Cloud Platform:

- **Access Policy Engine** to control the access to specific data.
- **Risk Management Monitor and Dashboard** to monitor and evaluate the risks related to the privacy exposure.
- **DataStream and Contract Composer** to manage the lifecycle of the contracts.
- **Policy-compliant Blockchain Infrastructure and DLT Engine** to facilitates the sealing of contracts on the side of the Individuals, as well as their compensation for assets that have been bought by Data Seekers.
- **Persona Generator** to support the data anonymization process.

Table of Contents

Introduction 7
1.1 Document structure 7
1.2 Relation to other WPs/Tasks 7
2 WP3 Components Descriptions – Personal App 8
2.1 Trusted Platform Module (TPM) Interface 9
2.1.1 Technology Background 9
2.1.2 Component Backlog 9
2.2 Sharing Configurator 11
2.2.1 Technology Background 12
2.2.2 Component Backlog 13
2.3 Data Request Service Resolver 17
2.3.1 Technology Background 19
2.3.2 Component Backlog 19
2.4 Personal Wallet 21
2.4.1 Technology Background 23
2.4.2 Component Backlog 23
2.5 Blockchain Security 2Go Starter Kit 25
2.5.1 Technology Background 26
2.5.2 Component Backlog 26
2.6 Data Anonymizer 27
2.6.1 Technology Background 29
2.6.2 Component Backlog 31
2.7 Attribute Based Encryption Engine 32
2.7.1 Technology Background 32
2.7.2 Component Backlog 32
2.8 Access Policies Editor 35
2.8.1 Technology Background 36
2.8.2 Component Backlog 36
3 WP3 Components Descriptions – Cloud Platform 37
3.1 Access Policy Engine 38
3.1.1 Technology Background 38
3.1.2 Component Backlog 39

3.2 Risk Management Monitor and Dashboard 41
3.2.1 Component Backlog 43
3.3 DataStream and Contract Composer 46
3.3.1 Component Backlog 49
3.4 Policy-compliant Blockchain Infrastructure and DLT Engine 51
3.4.1 Technology Background 52
3.4.2 Component Backlog 53
3.5 Persona Generator 55
3.5.1 Technology Background 56
3.5.2 Component Backlog 57
4 Conclusions and Next Steps 60

List of Figures

Figure 1: Sharing Configurator - Asset Selection for Sharing 11
Figure 2: Sharing Configurator - Anonymisation Selection 11
Figure 3: Sharing Configurator - Other Sharing Information 12
Figure 4: Sharing Configurator – Configuration Preview 12
Figure 5: View an open data sharing request 17
Figure 6: View of an example questionnaire 18
Figure 7: Blacklisting Data Seekers and Service Toggling 18
Figure 8: Personal Wallet main window 21
Figure 9: Personal Wallet, compensation details 22
Figure 10: Personal Wallet, merchant details 22
Figure 11: Personal Wallet, merchant purchase 23
Figure 12: User interface for interacting with the Blockchain Security 2Go Starter Kit 25
Figure 13: Anonymizer Data Analyst Page 28
Figure 14: Anonymizer Sharing Configurator Page 29
Figure 15: Workflow of Data Anonymizer 30
Figure 16: Mock-up of the Access Policy Editor page within the Data Sharing configurator 35
Figure 17: Mock-up of the privacy risk monitor 41
Figure 18: Mock-up of the privacy risk dashboard Technology Background 42
Figure 19: Mock-up of Acquiring a Dataset (from the Query Builder) 47
Figure 20: Mock-up of Requesting a Dataset 48
Figure 21: Mock-up of Building a Questionnaire Technology Background 48
Figure 22: DataVaults Quorum DLT Conceptual Architecture 51
Figure 23: DataVaults Multi-Tenant via multiple Private States Quorum Node 52
Figure 24: Persona Builder overview 56
Figure 25: Workflow of Persona Generator 57

Terms and Abbreviations

ABE	Attribute based encryption
API	Application Programming Interface
CIV	Configuration Integrity Verification
DAA	Direct Anonymous Attestation
DLT	Distributed Ledger Technology
NFC	Near Field Communication
REST	Representational State Transfer
TPM	Trusted Platform Module
UI	User Interface
UML	Unified Modelling Language
WP	Work Package

INTRODUCTION

This document represents an auxiliary document to deliver the code and give the implementation status of the components grouped under WP3. The theoretical foundation elaborated in WP2, in respect to technical, legal and ethical requirements, is put into practice with this work package.

1.1 DOCUMENT STRUCTURE

After the introduction in Section 0, the document continues with the description of the components of the DataVaults Personal App in Section 2 and the corresponding documentation of the DataVaults Cloud Platform in Section 3. It gives an overview and highlights the progress until M18 of the project runtime.

A short description of each component is enriched with mock-up illustrations or real-life screenshots (if it has a User Interface (UI) and depending on the development progress) to give an impression about the intended usage from the user's point of view. The technical details include the technology stack providing the foundation of the component.

For each component, the corresponding user stories coming out of WP5 are collected and a classification between already implemented and future extensions is performed. The features are described in the form "As a *<Role>*, I want to *<Action>*, so that *<Reason>*". This gives a clear indication about the required features from the individual stakeholders' views to guide the development. All user stories are collected in tables and assigned to components to record a backlog for additional features in upcoming releases.

Finally, a conclusion is given in Section 4.

1.2 RELATION TO OTHER WPS/TASKS

D3.1 is the first in a series of deliverables as part of WP3 activities, with close relations to WP5 and WP4. The overall system architecture with the classification in individual components, as well as the user stories describing the target functionalities, are directly integrated as input from WP5. Moreover, results from WP3 will be returned to WP5 activities for testing and integration into the overall DataVaults solution.

The DataVaults platform is developed in three self-contained iterations, extending the supported functionality with each step. This agile development approach, where the end-product is developed in consecutive iterations, allows for flexible consideration of new findings and design choices, which are also carried over to the architecture definition and user stories if changes are required.

Development of the adjoining components from WP3 and WP4 is done in parallel, causing a mutual dependency between both work packages. For this reason, the results of D3.1 and D4.1 will be directly incorporated in the next development phase of the counterpart.

2 WP3 Components Descriptions – Personal App

The DataVaults Personal App is the integrated application that resides at the end of the Data Owners and is used by these users to collect, store, process and share their data with Data Seekers. All the data is handled based on specific sharing configurations by the DataVaults Cloud-based Engine.

The majority of the components in the DataVaults Personal App are responsible for handling the data immediately after it is collected.

As such, this subsection provides the progress done in the following components of the DataVaults Personal App:

- Trusted Platform Module
- Sharing Configurator
- Data Request Service Resolver
- Personal Wallet
- Blockchain Security 2GO Starter Kit
- Data Anonymiser
- Attribute Based Encryption Engine
- Access Policies Editor

The source code of the different components which are open source, is provided in the following repository

https://www.gitlab.com/DataVaults

2.1 Trusted Platform Module (TPM) Interface

TPM is a low-level hardware component, which offers a variety of features to enhance the security and trust characteristics of an application. In DataVaults, it enables two advanced schemes by providing the required cryptographic primitives:

- Direct Anonymous Attestation (DAA): Authorization to the platform can be done in a privacy-preserving manner with pseudonyms. Nevertheless, access is restricted to pre-registered users and the platform can revoke the credentials of individual users.
- Configuration Integrity Verification (CIV): The validity of data from a user's device can only be ensured when the installed software and configurations are not altered to skip the security provisions. The TPM can certify the system state and only allow data from attested devices.

Both schemes require additional software installed and access to the TPM on the client device. For the browser-based access delivered as the initial alpha-release, this is not possible. Both features will only be available in upcoming releases with a standalone installer. Nevertheless, definition and implementation of the interfaces for seamless integration is done in parallel. The TPM interface does not have an exclusive user interface; it is instead part of other components (e.g. sharing configurator).

2.1.1 Technology Background

For the DAA and CIV schemes, we use already available implementations written in C++. Internally they rely on the IBM TPM Software Stack[1] (TSS) for interaction with the TPM API and OpenSSL for generic cryptographic operations.

The interface will be implemented as a local socket in *Python 3* to provide the API to other components, pre-process input data and forward calls to the responsible service.

2.1.2 Component Backlog

2.1.2.1 Features planned for upcoming Releases

ID#	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_01	Sharing Configurator. Sharing Setup Manager	Data Provider	Select if I want DAA to be used during data sharing	my data asset will have an extra "trustworthiness" guarantee.
US_68	Attestation	Data Provider	Authenticate my device in a privacy-preserving way	my data cannot be linked directly to my identity.

[1] https://sourceforge.net/projects/ibmtpm20tss/

US_69	Attestation	Data Provider	I want to get informed if my device supports DAA	I am aware of whether the strictest privacy-preserving mode is available for me
US_70	Attestation	Data Provider	choose an alternative authentication method, in case my device does not support DAA	I have compatibility with my devices and avoid technology lock-in.
US_71	Attestation	Data Provider	have the strictest privacy-preserving technology enabled by default	I have the best privacy preservation without further configuration.

2.2 Sharing Configurator

The sharing configuration is one of the core components of the DataVaults Personal App, as it is responsible for sharing the data of a Data Owner with interested stakeholders, by putting that data on display (setting different access/security and privacy policies) over the DataVaults Cloud Platform.

The visual interface of the Sharing Configurator as developed for the first release of the Alpha Version is shown in the next Figures.

Through this component, a user is able to select a data source (Figure 1) for sharing and provide extra details, then decide on whether it should be anonymized (and how) or not (see Section 2.6 and Figure 2).

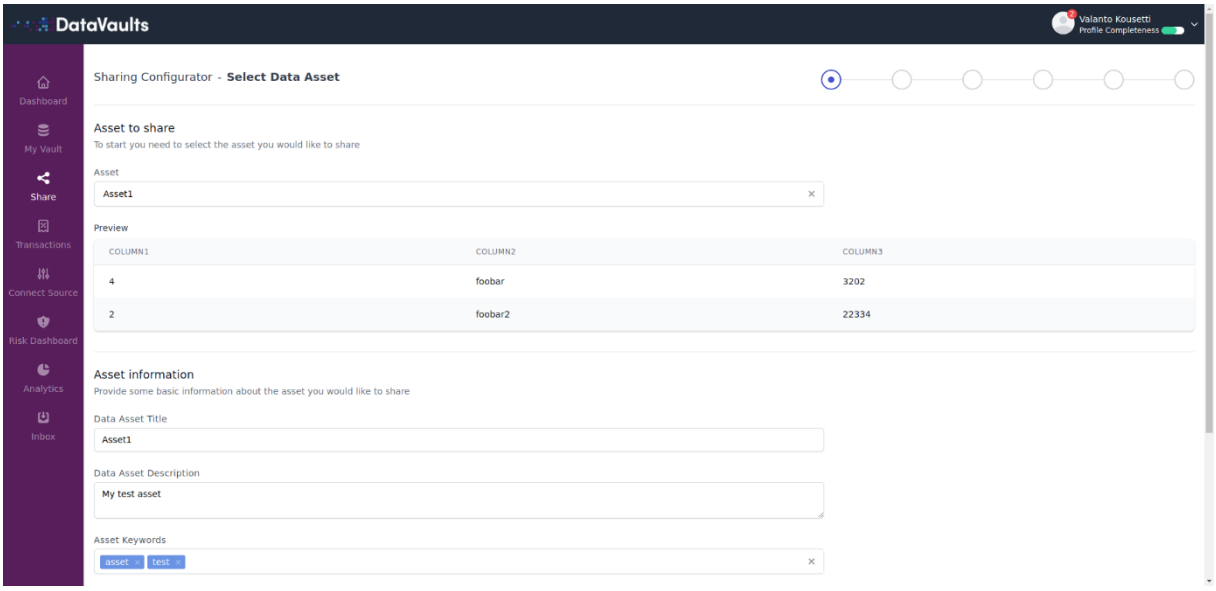

Figure 1: Sharing Configurator - Asset Selection for Sharing

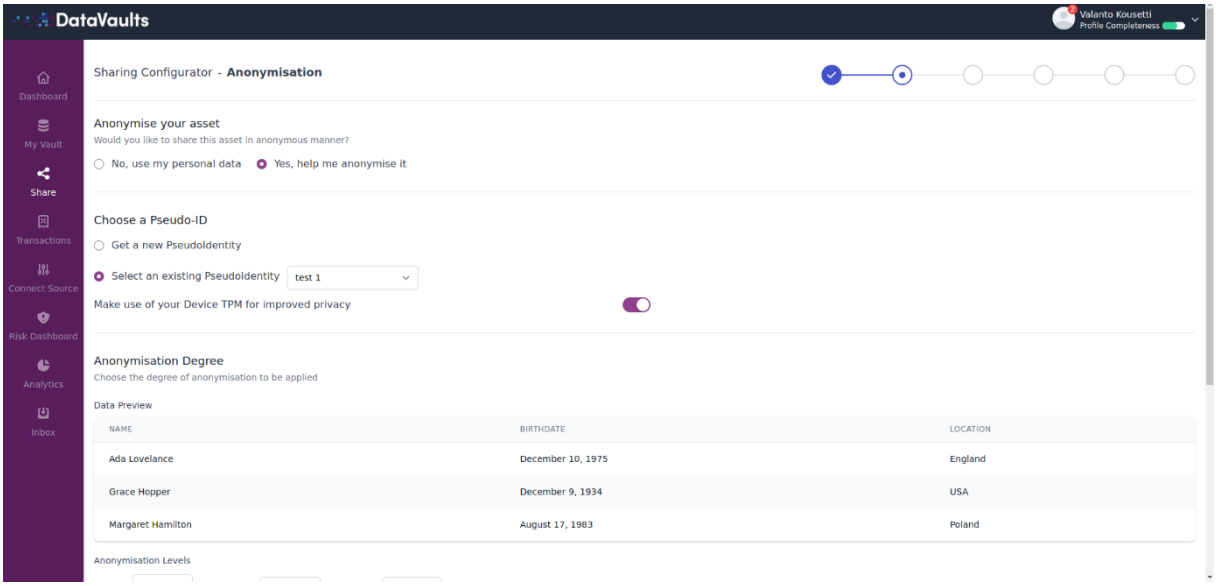

Figure 2: Sharing Configurator - Anonymisation Selection

The access policies for this specific data set are provided and finally details on pricing, license, etc. are defined before sharing (Figure 3).

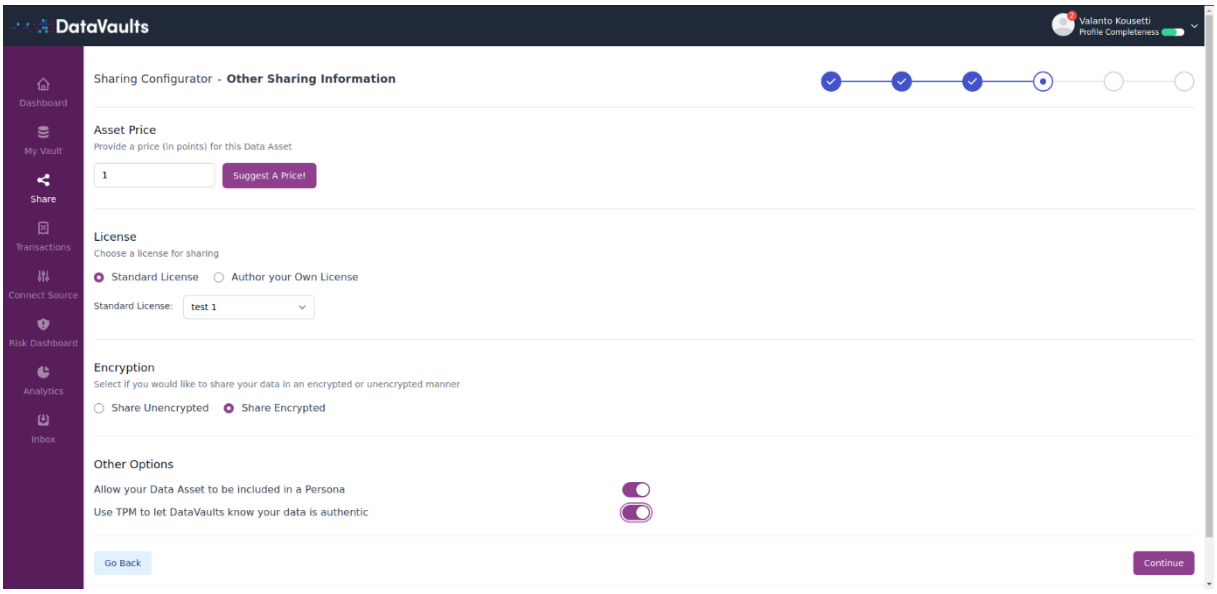

Figure 3: Sharing Configurator - Other Sharing Information

Finally, a summary of the selected configuration is displayed before the user can confirm the sharing operation.

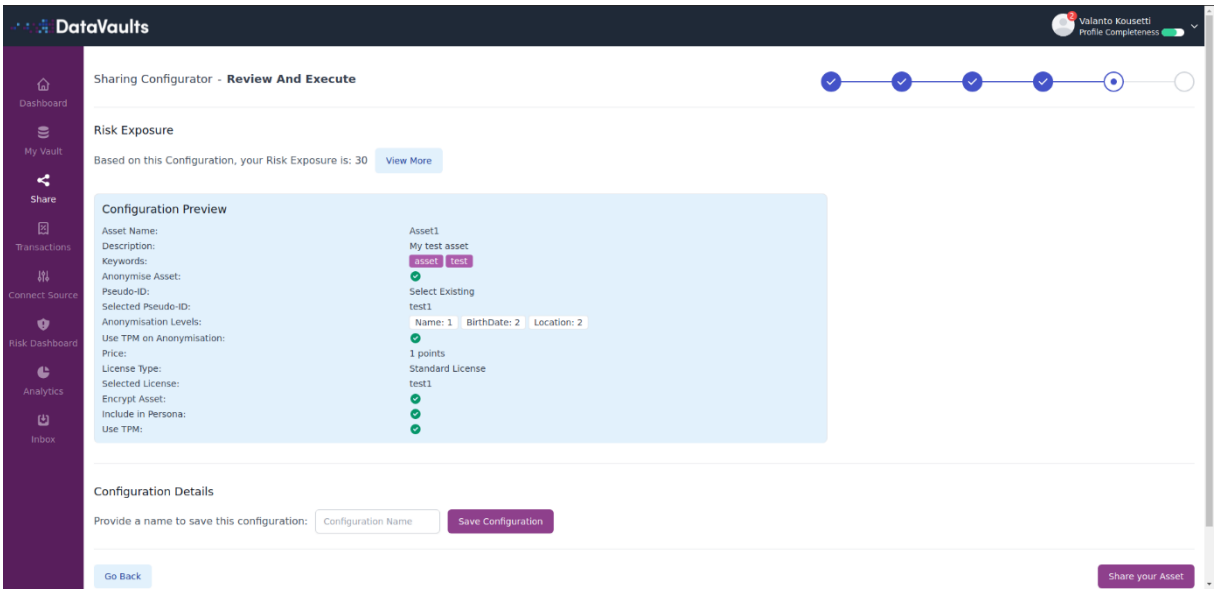

Figure 4: Sharing Configurator – Configuration Preview

2.2.1 Technology Background

Sharing Configurator is a component that is built with the VueJS 3 framework and is using information coming from other components, the Personal App's MongoDB and the Personal App's Postgres database.

2.2.2 Component Backlog

2.2.2.1 Implemented Features

ID#	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_01	Sharing Configurator. Sharing Setup Manager	Data Provider	create a new sharing configuration, with all fields empty	I can share/upload to the DataVaults Cloud a data asset
US_02	Sharing Configurator. Sharing Setup Manager	Data Provider	select inside the sharing configuration the data asset to be shared	I can make a copy of the selected data asset to the DataVaults Cloud for private or sharing purposes.
US_03	Sharing Configurator. Sharing Setup Manager	Data Provider	select inside the sharing configuration the data source to be shared	I can make copies of data collected from the selected data source to the DataVaults Cloud for private or sharing purposes.
US_04	Sharing Configurator. Sharing Setup Manager	Data Provider	select a data asset to be uploaded and/or shared to the DataVaults Cloud, from my Personal DataVaults App data assets	I can make a copy of the selected data asset to the DataVaults Cloud for private or sharing purposes.
US_05	Sharing Configurator. Sharing Setup Manager	Data Provider	select a connected data source to share data assets collected from there	I can make copies of data collected from this source to the DataVaults Cloud for private or sharing purposes.
US_06	Sharing Configurator. Sharing Setup Manager	Data Provider	select the anonymisation level of the selected data asset	I can choose what happens with the personally identifiable information that is available in the data asst that will be shared.
US_07	Sharing Configurator. Sharing Setup Manager	Data Provider	select the sharing level for the selected data asset (public, private - access policies in effect, confidential -not to be shared)	I can choose whether the specific data asset will be available to others or will only be uploaded for me.
US_08	Sharing Configurator. Sharing Setup Manager	Data Provider	select a time period for which the data asset will be available on the DataVaults Cloud for sharing	I control the availability of my data.

US_09	Sharing Configurator. Sharing Setup Manager	Data Provider	select what part of the data asset and its metadata wil be publicly available for preview purposes	I control what is available from my data and at the same time higher the chances of my data being purchased by Data Seekers.
US_10	Sharing Configurator. Sharing Setup Manager	Data Provider	select from a list of predefined licences that should be in effect for the shared data, regarding usage (e.g., no distribution, sharing with reference to source etc.), time (e.g., one-month, one year, forever)	I control that my data assets that are shared are used properly.
US_11	Sharing Configurator. Sharing Setup Manager	Data Provider	edit the fields of the licence terms	I can customise the various licence parameters.
US_12	Sharing Configurator. Sharing Setup Manager	Data Provider	view a suggested price for my data asset under the selected sharing options	I can decide easier on a price that maximises my earnings, while remaining competitive in the personal data market ecosystem.
US_13	Sharing Configurator. Sharing Setup Manager	Data Provider	select a price tag for the data asset, under the selected sharing configuration (licence, anonymisation level etc.)	I can be compensated whenever the data asset is acquired by a Data Seeker.
US_15	Sharing Configurator. Sharing Setup Manager	Data Provider	execute the configured sharing	my data asset is uploaded to the DataVaults Cloud Platform.
US_22	Sharing Configurator. Sharing Setup Manager	Data Provider	load a saved sharing configuration for a new data asset to be shared	I can reuse an existing configuration and make any adaptations needed for the new data asset to be shared.
US_27	Access Policy Editor	DataVaults Personal App	Receive the identification of the Individual and load the access policies on the Access Policy Editor interface	The Individual can configure the policies for granting access to her data.
US_28	Access Policy Editor	Data Provider	edit the access policies that apply to my data assets	I change the terms for providing access to my data
US_29	Access Policy Editor	Data Provider	load existing access policy templates for creating new policies	I can easily define the access policies that will apply to my data.

US_30	Access Policy Editor	Data Provider	create new templates from my policies	I can re-use it in the future.
US_31	Access Policy Editor	Data Provider	finalise the policies configuration of a data sharing configuration.	these policies take effect once the sharing configuration is executed.

2.2.2.2 Features planned for upcoming Releases

ID#	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_14	Sharing Configurator. Sharing Setup Manager	Data Provider	Select if I want DAA to be used during data sharing	my data asset will have an extra “trustworthiness” guarantee.
US_16	Sharing Configurator. Sharing Setup Manager	Data Provider	Modify the sharing parameters of a data asset I have already shared	my data asset is from this point shared under the new terms.
US_17	Sharing Configurator. Sharing Setup Manager	Data Provider	save the sharing configuration at any stage	I can return to it at a later stage to continue or reload it for reuse.
US_18	Sharing Configurator. Sharing Setup Manager	Data Provider	create a sharing schedule using a saved sharing configuration for a connected data source	my data assets are shared in an automated way and I don't have to reconfigure the sharing process again and again.
US_19	Sharing Configurator. Sharing Setup Manager	Data Provider	pause a sharing schedule	I freeze the sharing of specific data assets for a while, until I decide to resume.
US_20	Sharing Configurator. Sharing Setup Manager	Data Provider	delete a sharing schedule	the specific sharing configuration is no longer a recurring event.
US_21	Sharing Configurator. Sharing Setup Manager	Data Provider	delete a sharing configuration	I discard any sharing configuration I no longer wish to use.
US_22	Sharing Configurator. Sharing Setup Manager	Data Provider	load a saved sharing configuration for a new data asset to be shared	I can reuse an existing configuration and make any adaptations needed for the new data asset to be shared.
US_23	Attestation	Data Provider	Authenticate my device in a privacy-preserving way	my data cannot be linked directly to my identity.

US_24	Attestation	Data Provider	I want to get informed if my device supports DAA	I am aware of whether the strictest privacy-preserving mode is available for me
US_25	Attestation	Data Provider	choose an alternative authentication method, in case my device does not support DAA	I have compatibility with my devices and avoid technology lock-in.
US_26	Attestation	Data Provider	have the strictest privacy-preserving technology enabled by default	I have the best privacy preservation without further configuration.

2.3 DATA REQUEST SERVICE RESOLVER

The Data Request Service Resolver is the component responsible for getting a data sharing request coming from a Data Seeker and translating this to a sharing configuration proposition to display to the Data Owner, who has the final say regarding accepting or rejecting such a contract.

As such, the data request is presented to the Data Owner, alongside with a message from the Data Seeker as shown in the next figures. The relevant information is displayed in a clear way, to allow users of the target audience to identify the requested asset and understand the implications of the sharing activity. They have the option to accept this request or reject it.

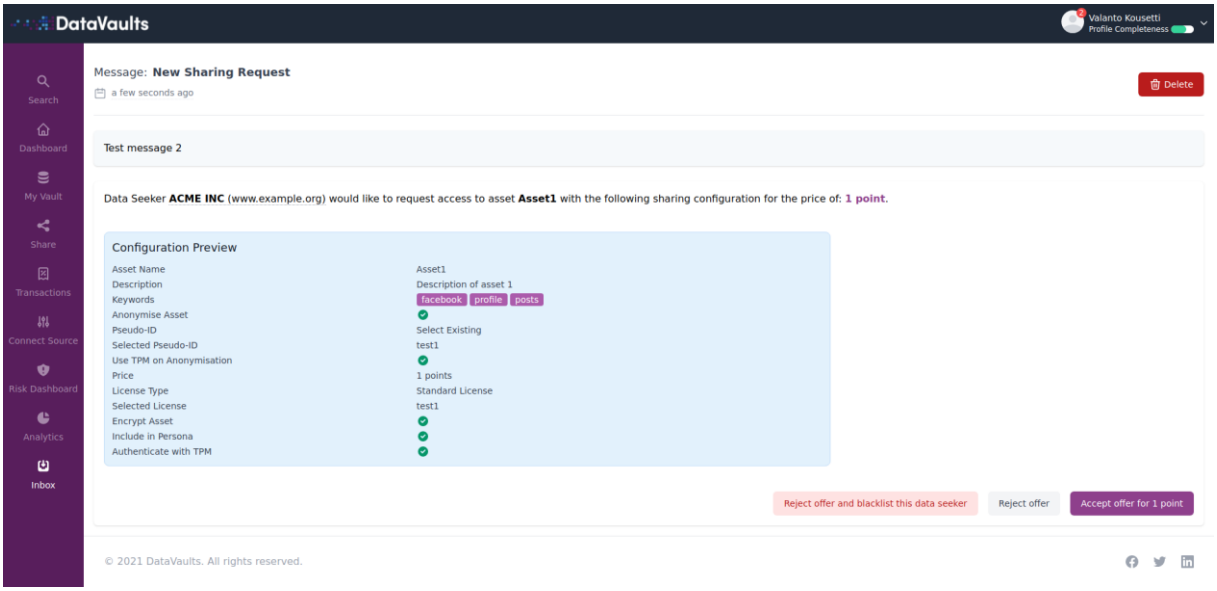

Figure 5: View an open data sharing request

In case the request contains a questionnaire, then this component renders the questionnaire at the side of the user, as illustrated in Figure 6.

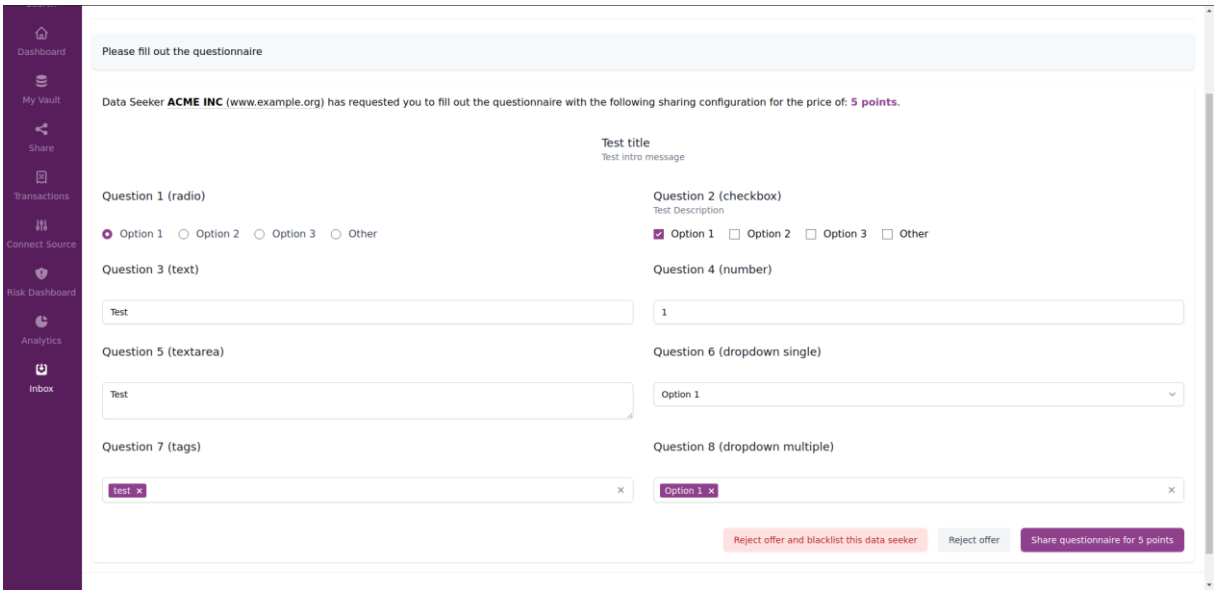

Figure 6: View of an example questionnaire

To mitigate the occurrence of unwanted data requests, the Personal App can be configured to automatically reject all requests from certain Data Seekers.

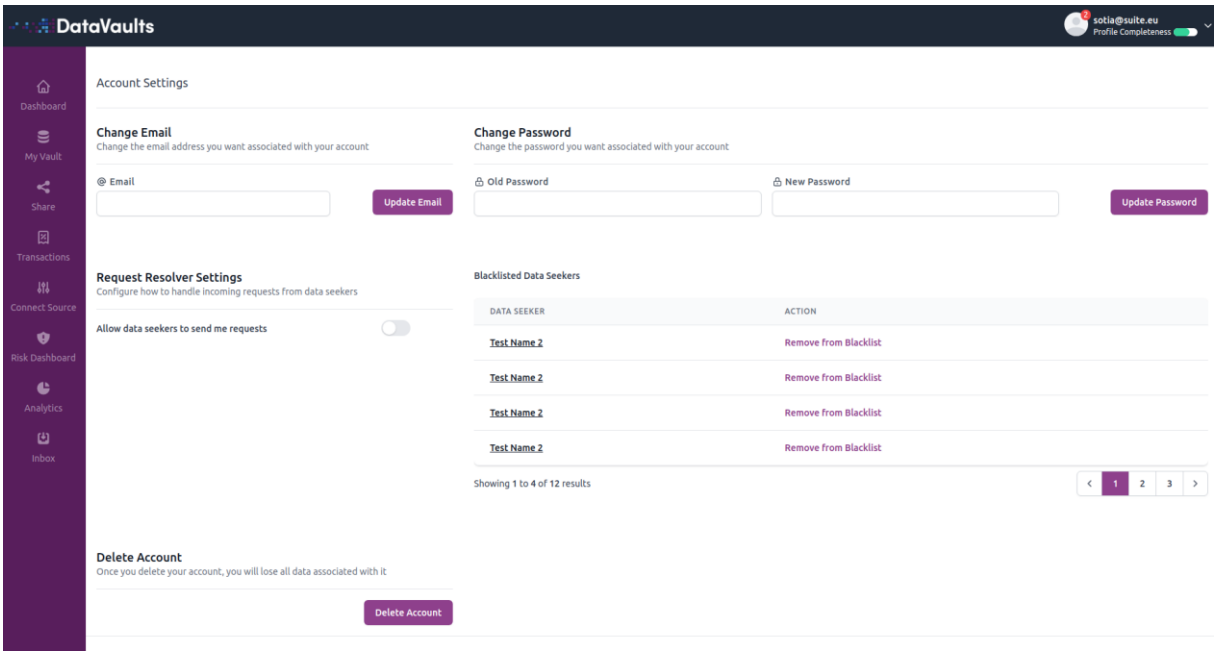

Figure 7: Blacklisting Data Seekers and Service Toggling

2.3.1 Technology Background

The Data Request Service Resolver is built with the VueJS 3 framework and uses information coming from other components, the Personal App's MongoDB and the Personal App's Postgres database. The messaging protocol for getting such requests from the cloud-based infrastructure is based on RabbitMQ.

2.3.2 Component Backlog

2.3.2.1 Implemented Features

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_100	Data Request Service Resolver	Data Provider	receive a notification on my Personal DataVaults App whenever a custom request for my data is made	I am instantly informed of any pending requests.
US_101	Data Request Service Resolver	Data Provider	view the details of the sharing proposal made by the Data Seeker, including the requested type of data, usage, licence, price, seeker's information (organisation etc.).	I have a full overview of the sharing & usage terms prior to accepting or rejecting the proposal.
US_102	Data Request Service Resolver	Data Provider	accept the sharing proposal/request	the sharing of my data asset under the accepted terms can take place.
US_103	Data Request Service Resolver	Data Provider	reject the sharing proposal/request	I can keep having some data only on the personal DataVaults side and not share them with anyone else.
US_104	Data Request Service Resolver	Data Provider	modify the notification settings	I can block recurrent requests without disabling the service completely.

US_105	Data Request Service Resolver	Data Provider	disable the service completely	I will not get any notifications in the future.
US_106	Data Request Service Resolver	Data Provider	blacklist certain Data Seekers or business sectors from performing requests	I quickly filter out Data Seekers with whom I am not interested to share data

2.3.2.2 Features planned for upcoming releases

All identified features of the initial platform requirements have been delivered. If new user stories are identified during development and demonstrator validation (WP6), they will be included in subsequent deliverables.

2.4 PERSONAL WALLET

The personal wallet module provides the functionality required by Data Owners to receive and use compensations they receive as result of sharing data while keeping privacy of users during the whole flow of compensations, since they are received and transferred to the wallet until they are used.

The wallet also provides an interface for merchants to offer products to be acquired, or services used with compensations. This is enabled by publishing products and performing the compensation exchange for products or services through the DataVaults backend services.

On the main window of the Wallet component, the user can choose to see the details of completed transactions or go to the merchand pages.

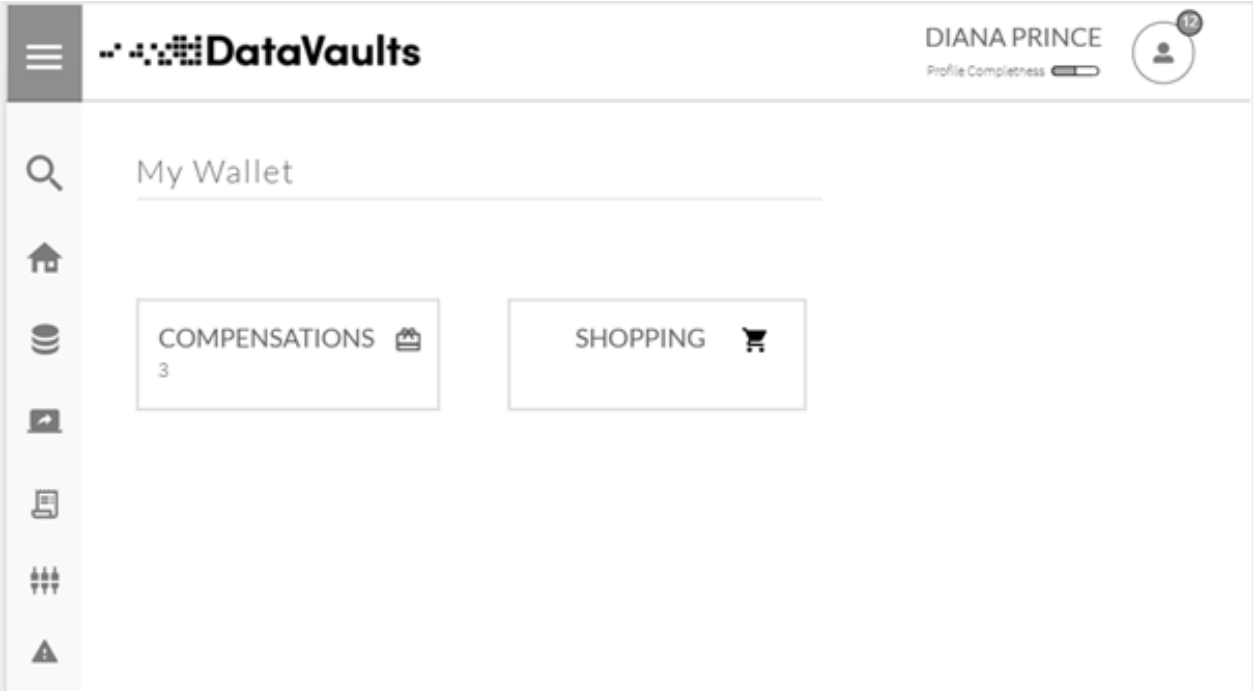

Figure 8: Personal Wallet main window

The compensation details give information about the status, amount and source of the compensation.

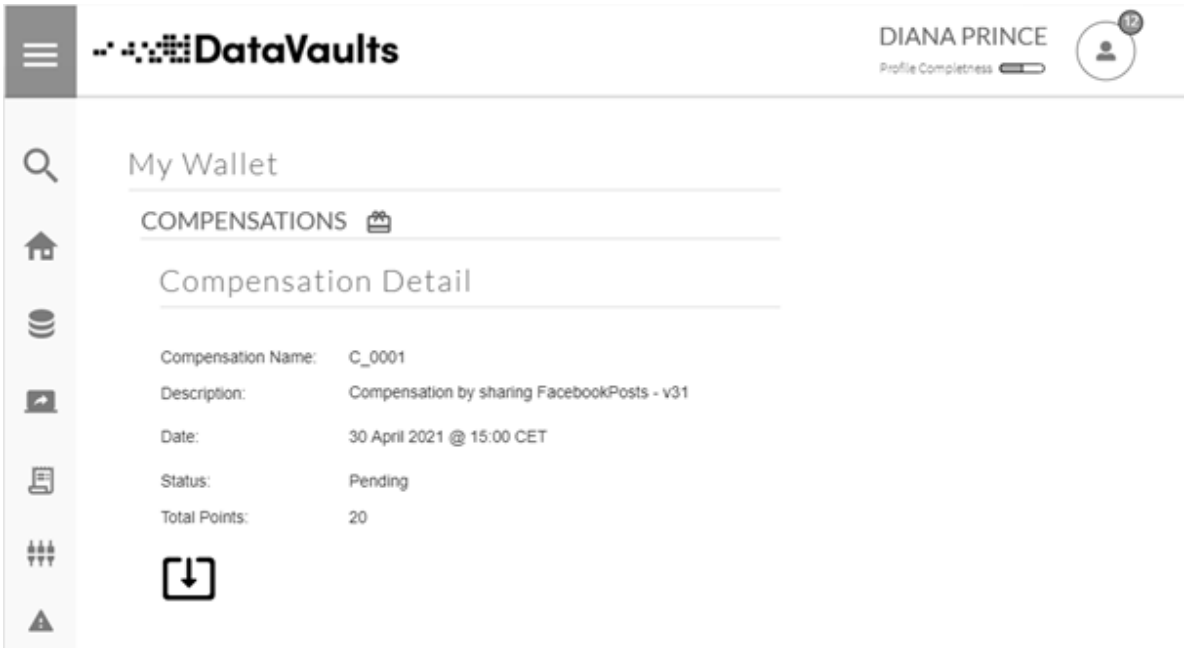

Figure 9: Personal Wallet, compensation details

The Personal Wallet component will enable multiple merchants to offer their goods and services. The user can then exchange previously received compensation for a range of products (Figure 11).

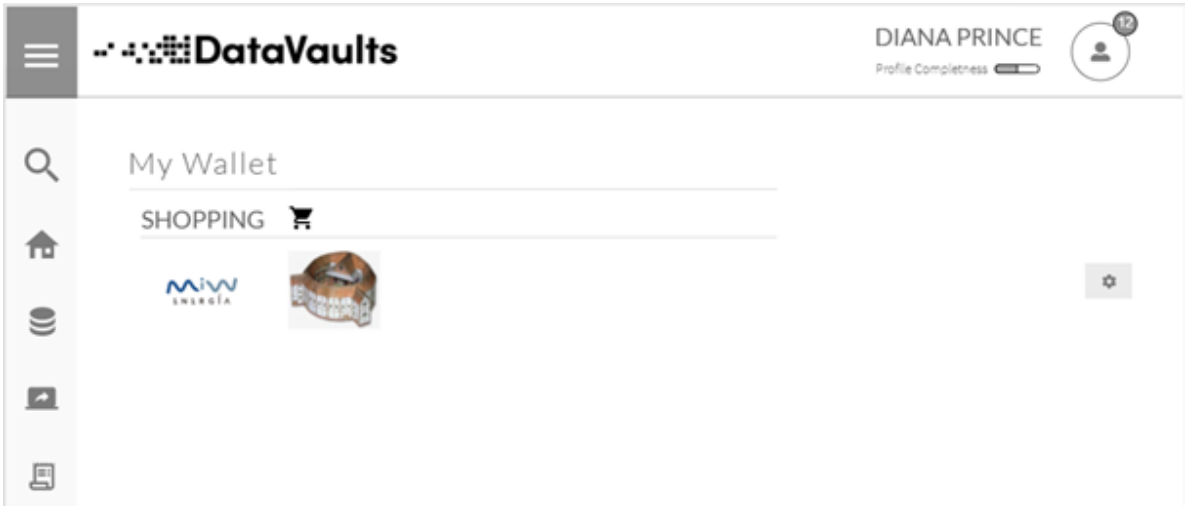

Figure 10: Personal Wallet, merchant details

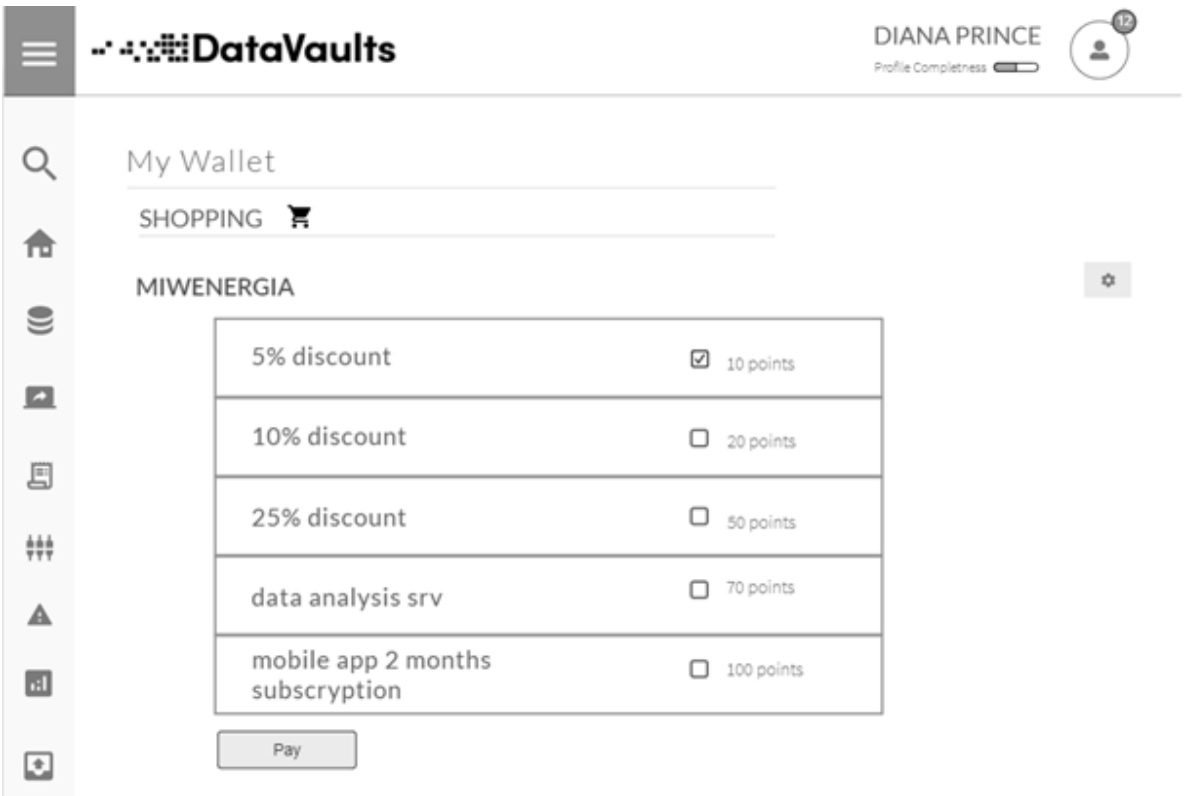

Figure 11: Personal Wallet, merchant purchase

2.4.1 Technology Background

The personal wallet is based on the development carried out in the *Functional Encryption TEChnologies* (FENTEC) project for the Privacy Enhanced Digital Currency Prototype[2]. This tool will be integrated with the Blockchain Security2Go Starter Kit to improve the security of the encryption scheme related to the creation of crypto tokens and their usage.

2.4.2 Component Backlog

2.4.2.1 Implemented Features

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_194	Personal DataVaults Wallet	Data Provider	use multiple blockchain addresses	I keep the data from unrelated sources separate.

[2] http://fentec.eu/sites/default/files/fentec/public/content-files/deliverables/FENTEC_D7.5_v1.0.pdf

US_196	Personal DataVaults Wallet	Data Provider	display the value of all accounts in my wallet	I can evaluate the worth of the previous sharing activities.
US_198	Personal DataVaults Wallet	Data Provider	exchange the funds into real-world goods or money	a real value is obtained from the data.
US_199	Personal DataVaults Wallet	Data Provider	securely store the private keys for my accounts	they cannot be leaked by other applications on the device.
US_200	Notification System	Data Provider	have a clear indication when my private key is used for signing a Blockchain transaction	no immutable actions are performed accidentally.
US_201	Personal DataVaults Wallet	Data Provider	spend my earnings, preserving my anonymity	the use of compensations does not leak information about me

2.4.2.2 Features planned for upcoming Releases

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_195	Personal DataVaults Wallet	Data Provider	see a list of all my blockchain addresses	I can view my previously used addresses.
US_197	Personal DataVaults Wallet	Data Provider	create a new account	a new pseudonym is used for subsequent data uploads.

2.5 Blockchain Security 2Go Starter Kit

The Blockchain Security 2Go Starter Kit is a smart card with Near Field Communication (NFC) interface for securely managing keys and creating signatures. It is optionally used to sign the Blockchain transactions from the user, e.g. new sharing configurations. In DataVaults it is additionally utilized to manage key pairs for the previously described Personal Wallet component.

When a new sharing configuration is created, or a payment is started, a signature request is forwarded to a program installed on the user's device. It will display a notification about the new request, show a control value and ask the user to hold the card to the reader. The signature is then returned to the Personal App, which then forwards it to the cloud backend.

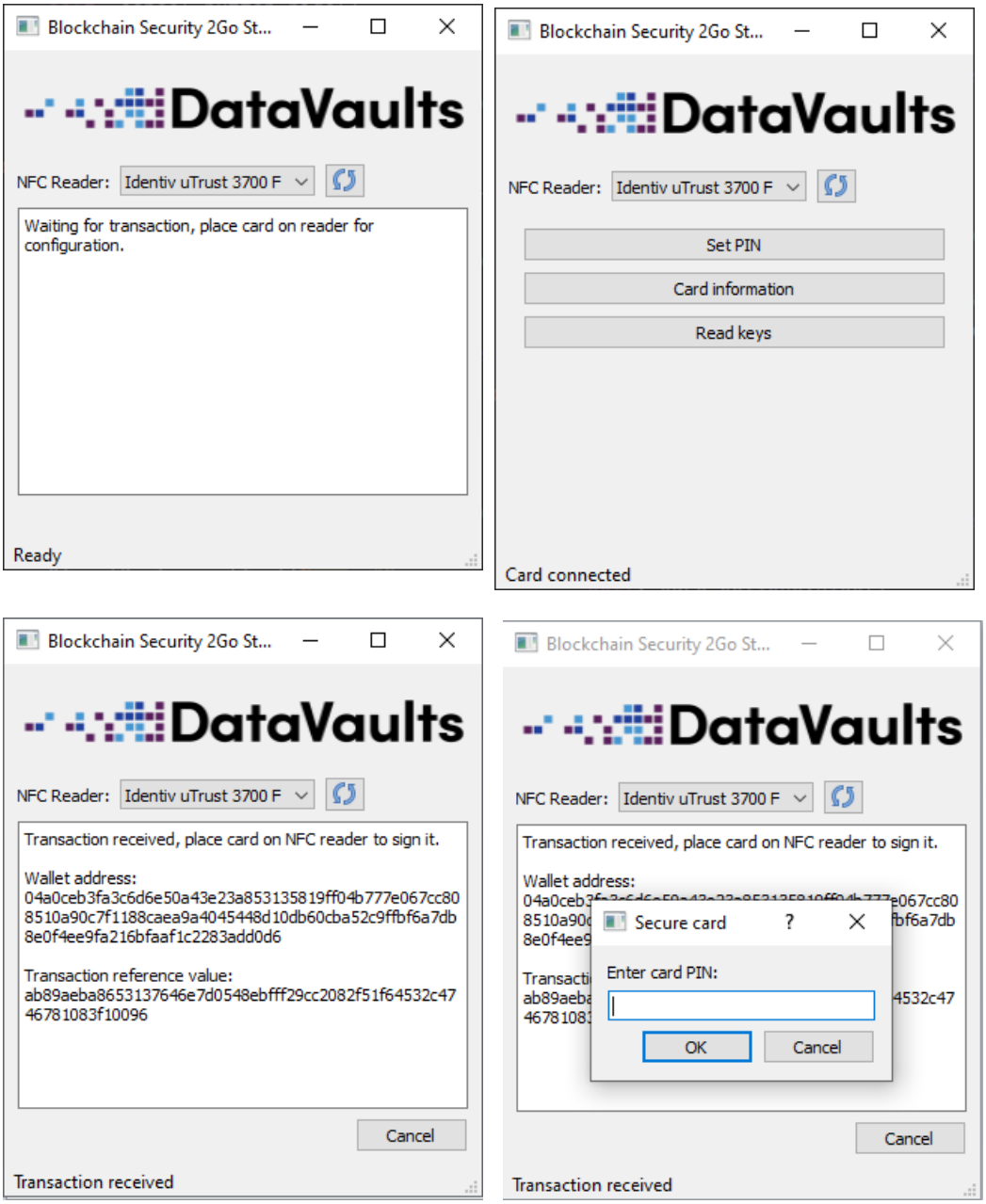

Figure 12: User interface for interacting with the Blockchain Security 2Go Starter Kit

2.5.1 Technology Background

A locally installed program forwards the communication from the web browser to the Blockchain Security 2Go Starter Kit through an NFC-reader. This software, called *Bridge*, listens for requests from the DataVaults Personal App with a local web server and initializes the low-level interface to the NFC reader.

The *Bridge* is written in Python and uses the pyscard and blocksec2go[3] libraries for the communication with the smart card. A Flask server is providing the API for the Personal App, PySide2 is included as GUI framework.

2.5.2 Component Backlog

2.5.2.1 Implemented Features

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_199	Personal DataVaults Wallet	Data Provider	securely store the private keys for my accounts	they cannot be leaked by other applications on the device.
US_200	Notification System	Data Provider	have a clear indication when my private key is used for signing a Blockchain transaction	no immutable actions are performed accidentally.

2.5.2.2 Features planned for upcoming Releases

All identified features of the initial platform requirements have been delivered. If new user stories are identified during development and demonstrator validation (WP6), they will be included in subsequent deliverables. The UI design is considered a functional prototype and will also be extended further.

[3] https://github.com/Infineon/BlockchainSecurity2Go-Python-Library

2.6 DATA ANONYMIZER

The anonymizer is the component of DataVaults responsible for preserving data privacy. It alters the data in such a way, that it will preserve its usefulness but hide the original data. With this modifications, it cannot be traced back to the individuals the data was taken from.

The anonymizer is capable of taking a dataset and obfuscating the contained data by replacing it with values that represent the original data in a way that is non-identifying (e.g. an age of 29 may be replaced with [20-30] or a name Darren Smith may be replaced with Darren *****).

Via the frontend interface, users will be able to configure the anonymization process by selecting different anonymization pre-sets which can be applied to a column in their dataset or they can use advanced settings to allow for more configurability in their anonymization. To see how their choices will impact the final result, a preview button is available. Upon clicking this button, users will see a subset of their dataset with their current anonymization options applied to it. This will help users understand the impact of their choices.

The PseudoID generator is a smaller component, capable of producing a unique ID for a user who wishes to share their data as an anonymous user. This ID may then be used for the purposes of communication with the data owner whilst preserving their anonymity.

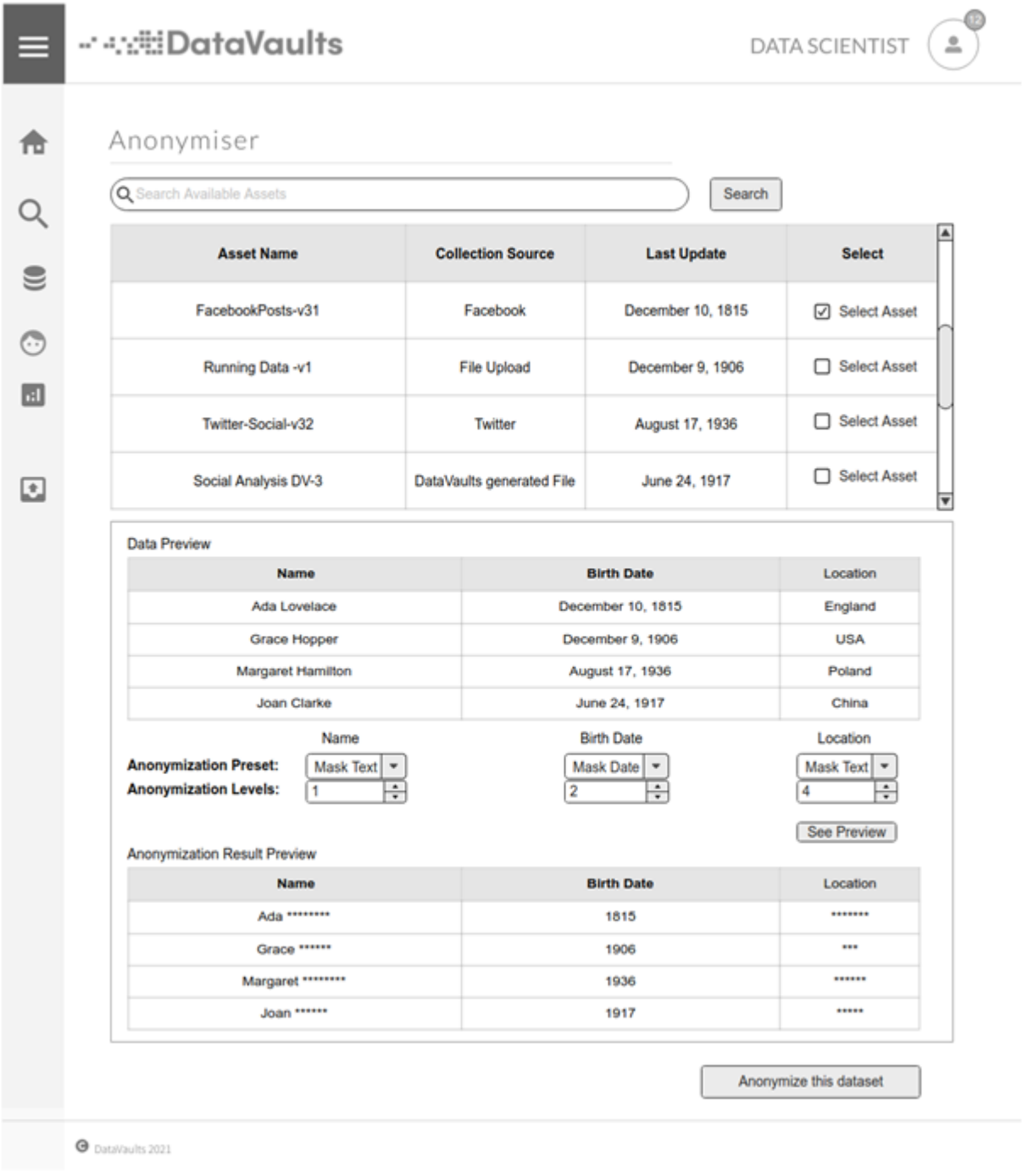

Figure 13: Anonymizer Data Analyst Page

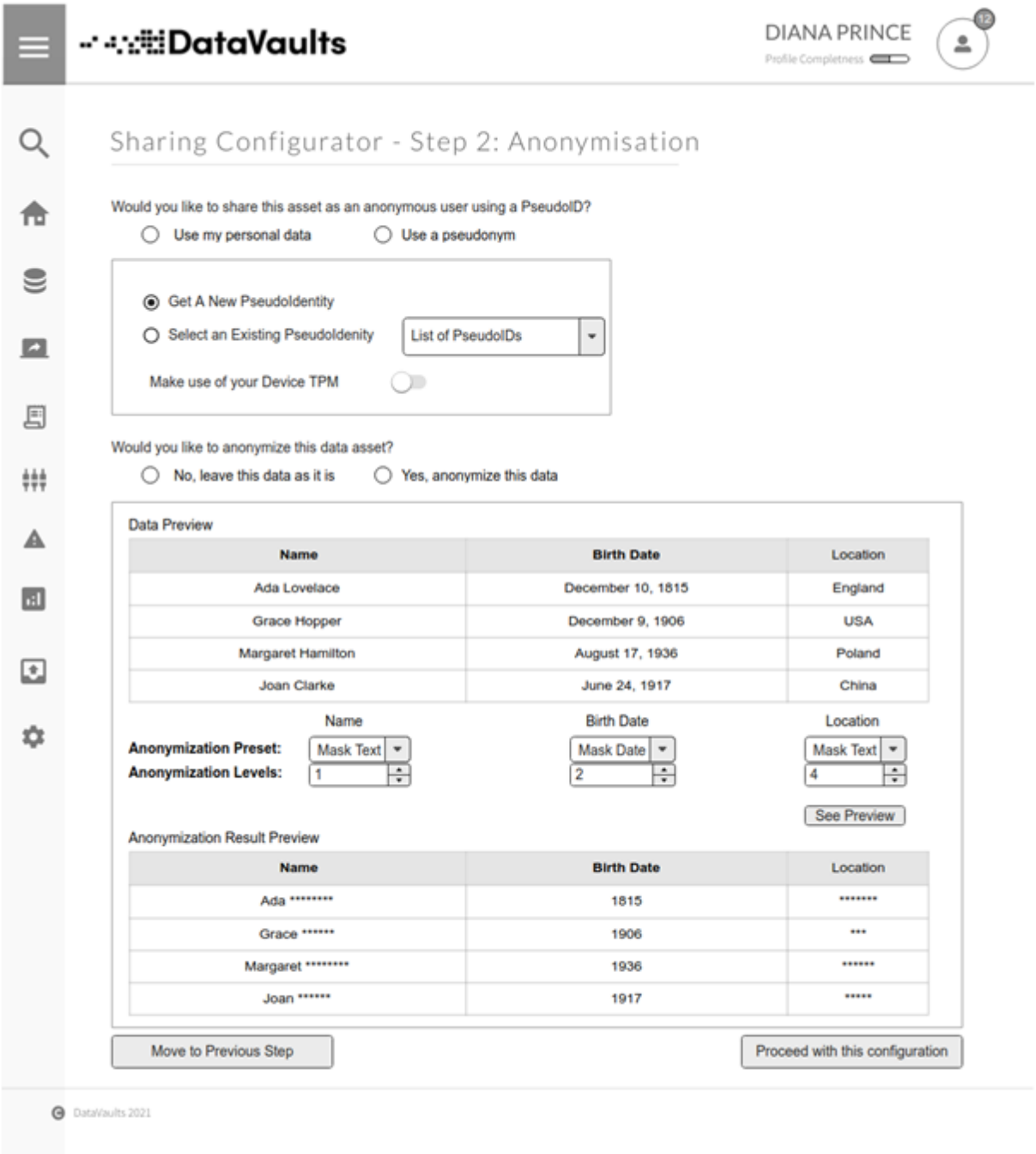

Figure 14: Anonymizer Sharing Configurator Page

2.6.1 Technology Background

Currently, the anonymizer takes the form of a Java application with 3 classes: AnonHandling contains methods responsible for configuring and executing the anonymization process; BuildTableHandling contains methods responsible for finding and returning the HierarchyBuilders to the AnonHandling methods, so that they can be used when configuring the anonymization process; and HierBuilding contains methods responsible for generating

new HierarchyBuilders, although these methods are currently unused, as these will only become relevant once users are allowed to generate their own HierarchyBuilders.

The final desired workflow for the anonymizer application is documented in the following UML diagram:

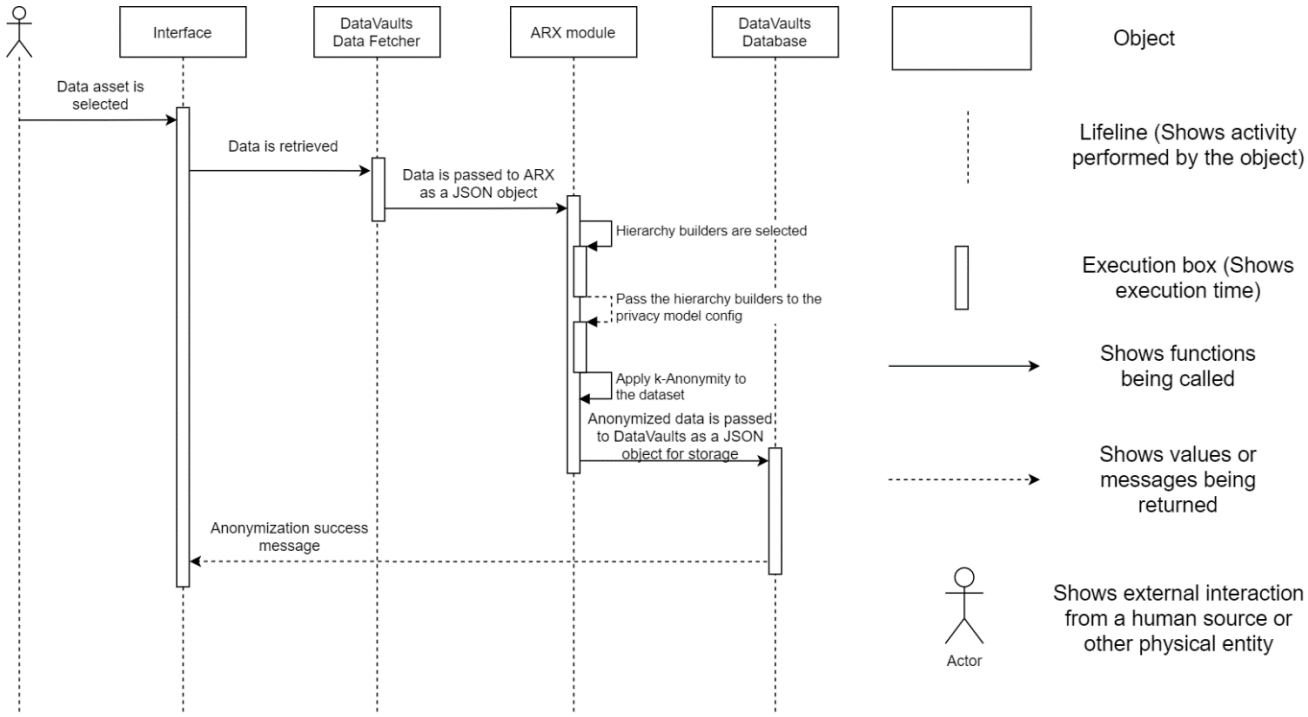

Figure 15: Workflow of Data Anonymizer

Using the hierarchy builder enables the creation of hierarchies on the fly as opposed to utilising pre-defined generated hierarchies that could prove restrictive if new data sources are not supported by the library of hierarchies available. The hierarchy builder will enable the anonymizer to create hierarchies of the following types:

- Redaction based hierarchies
- Interval based hierarchies
- Order based hierarchies
- Date based hierarchies

These take the form of a locally stored *ahs* file that stores information containing the rules required to build a hierarchy during runtime. By importing these hierarchy builders when we need them, we avoid the need to keep hierarchies stored and update them whenever we encounter new values.

2.6.2 Component Backlog

2.6.2.1 Implemented Features

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_226	Data Anonymiser	Data Provider	apply anonymisation to my data	personal information can be hidden
US_227	Data Anonymiser	Data Provider	create a fake ID	to hide my real ID that my data belongs to
US_228	Data Anonymiser	Data Provider	be able to select a pre-set anonymisation level	I can anonymise data using a specific approach

2.6.2.2 Features planned for upcoming Releases

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_229	Data Anonymiser	Data Provider	be able to configure the anonymisation approach used on my data	I have flexibility over how my data is shared

2.7 Attribute Based Encryption Engine

The ABE engine is one of the modules in DataVaults which implement sharing data preferences of data owners by providing access management to encrypted data.

This component provides data owners to set access policies to data sets or documents as a whole or by parts. This enables the application of different encryption patterns to data. As it is defined in D2.3, the ABE engine uses patterns to describe the encryption behaviour, to enable users to encrypt a document as a whole with a single access policy or encrypt the same document splitting it in pieces and applying a different access policy to each.

For the current version of the engine, a graphical interface has not been yet defined. Therefore in this first release the ABE engine will use a set of predefined policies to validate the encryption by patterns approach. It will be considered to use the same policy as the one defined in the Policy Editor module. Translation of this policy to the format used in the ABE engine is a requirement, as this is a Boolean expression format which entails more expressiveness limitations.

Key management will be simplified to one Master and public key pair for all participants; therefore, there is no need to implement a secure key hosting service for ABE at this stage.

2.7.1 Technology Background

The ABE engine is formed by four components implemented in JAVA and based on the ABE encryption schemes developed in the *Functional Encryption TEChnologies* (FENTEC)[4] project. These components are aimed to cope with each of the next functionalities: Key management, policy management, encryption and decryption.

2.7.2 Component Backlog

2.7.2.1 Implemented Features

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_90	ABE Engine	Data Provider	protect specific pieces of data	they can be access only by those who I give permission to
US_91	ABE Engine	Data Provider	I want to decide who can access my encrypted data	so that I choose who can learn what about me

[4] https://fentec.eu/

US_92	ABE Engine	Data Provider	apply different access policies to each piece of data through attribute encryption	that I can define cumulative access levels to my information
US_98	ABE Engine	Data Seeker	simplify the access to encrypted data as much as possible	so that I do not need to perform extra operations.
US_99	ABE Engine	DataVaults Cloud Platform	I want to be able to decrypt the data using ABE on behalf of a Data Seeker who has purchased it	I provide the data to the Data Seeker

2.7.2.2 Features planned for upcoming Releases

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_93	ABE Engine	Data Provider	I want to manage how queries are done over my encrypted data	I have extra level of protection against data correlation
US_94	ABE Engine	Data Provider	I want to be able to revoke/modify access to my encrypted data	I can re-define access policies applied to my encrypted data
US_95	ABE Engine	Data Provider	some data being accessible only from my personal App	the platform let me differentiate between data to be shared and operational data.
US_96	ABE Engine	DataVaults Cloud Platform	improve the performance on searching operations	to avoid searching over protected data will safe resources

US_97	ABE Engine	Data Seeker	I want to have access to aggregated data which I could not have in case the subject of these data was identifiable to	to obtain more expressive of richer data sets

2.8 Access Policies Editor

The Access Policies Editor is envisioned as part of the Data Sharing configuration in DataVaults. It allows the individuals to define the conditions under which their data will be shared.

The editor shows the current policy for the selected Data set if it already has any previously assigned. The individual can modify it, create a new one or load one of the locally stored policies for reusing them.

The attributes of the seekers are presented to choose the values for granting access to the data, allowing to select more than one value per attribute. These attributes can be seen in Figure 16.

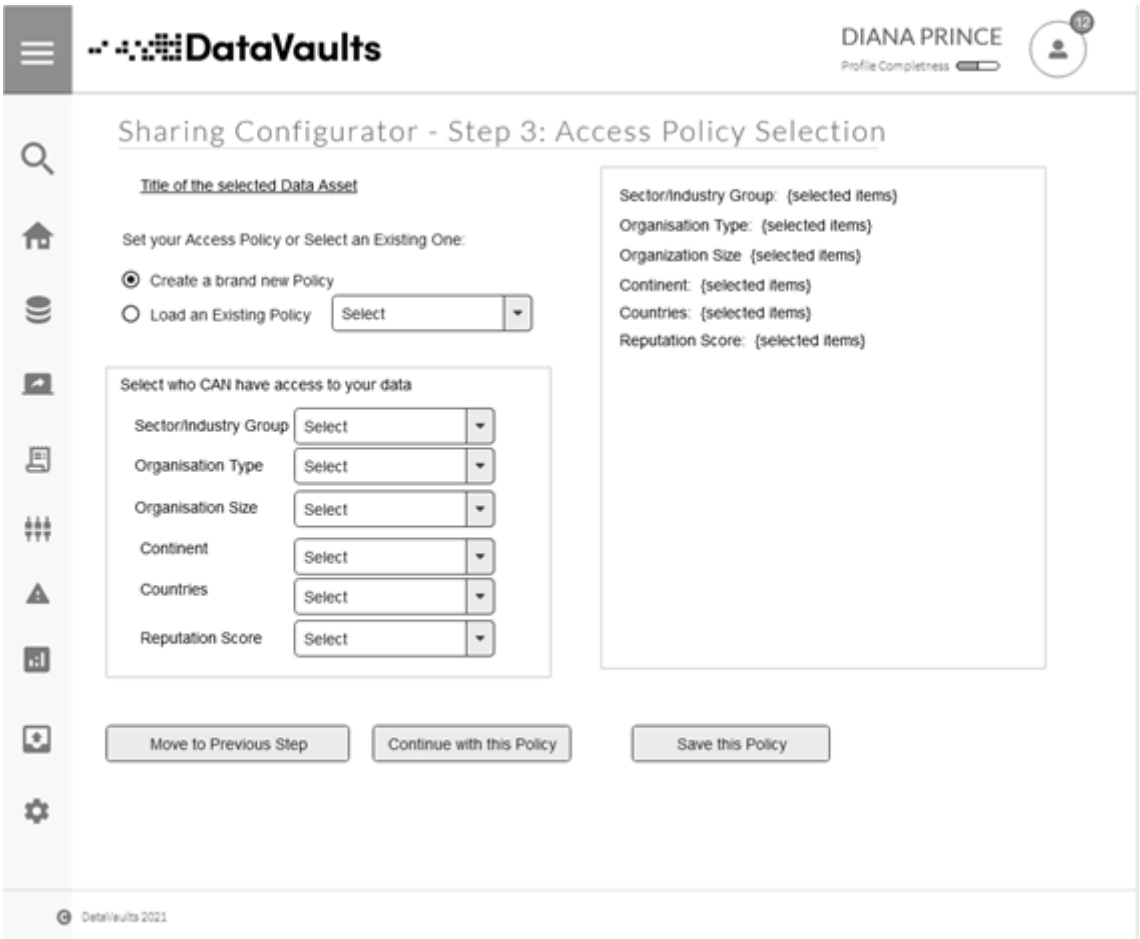

Figure 16: Mock-up of the Access Policy Editor page within the Data Sharing configurator

The main functionalities provided by this component are:

- Create a new brand policy regarding the previously selected Data set.
- Load an existing policy from a local repository. These policies are a set of reusable policies stored by the individual.
- Select the values for allowing access from a list of attributes of the seekers. The selection made appears on the right side of the page when accepting it.
- Save the conditions established in the current page locally as a reusable policy.
- Confirm the policy as the valid one and continue with the rest of the data sharing configuration.

2.8.1 Technology Background

As part of the Sharing Configurator component, it is built with the same VueJS3 framework and uses information coming from previous pages in this configurator.

2.8.2 Component Backlog

2.8.2.1 Implemented Features

ID#	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_27	Access Policy Editor	DataVaults Personal App	Receive the identification of the Individual data set and load the access policies on the Access Policy Editor interface	The Individual can configure the policies for granting access to her data.
US_28	Access Policy Editor	Data Provider	load existing access policy templates for creating new policies	I can easily define the access policies that will apply to my data.
US_29	Access Policy Editor	Data Provider	create new templates from my policies	I can re-use it in the future.
US_30	Access Policy Editor	Data Provider	edit the access policies that apply to my data assets	I change the terms for providing access to my data
US_31	Access Policy Editor	Data Provider	finalise the policies configuration of a data sharing configuration.	these policies take effect once the sharing configuration is executed.

2.8.2.2 Features planned for upcoming Releases

All the features are implemented. However, an evolution of the features is planned. The list of attributes will be analysed and the scope of these features extended if needed. Possible additions may arise from the demonstrator validation in WP6.

3 WP3 Components Descriptions – Cloud Platform

The DataVaults Cloud Platform is a cloud service offering a single-entry point for Data Seekers. It collects includes the cloud-based infrastructure, and from the WP3 perspective, it includes the backend and frontend part of the following components:

- **Access Policy Engine** to control the access to specific data;
- **Persona Generator** to support data anonymization process;
- **Risk Management Monitor** to monitor and evaluate the risks related to the privacy exposure;
- **Data Stream & Contract Composer** to manage the lifecycle of the contracts;
- **Trusted DLT Engine and the Public and Private Ledgers** to facilitates the sealing of contracts on the side of the Individuals, as well as their compensation for assets that have been bought by Data Seekers;

The source code of the different components, which are open source, is provided in the following repository

https://www.gitlab.com/DataVaults

3.1 Access Policy Engine

The Access Policy Engine is part of the DataVaults platform and responsible for analysing if a request of accessing data, made by the Data Seekers or by another component in the DataVaults platform, is going to be granted.

The process followed by the Engine consists of comparing the current values of the attributes informed by Data Seekers and the values established as allowed by the data owners when configuring the Access Policies.

Once the Engine has performed the decision process, the response given from the component will consist of a Boolean parameter "granted", and in addition, if the access is not granted (false), a list of non-conformities will be sent to the caller. That outcome can be useful in case the Data Seeker wants to change the request or the attributes informed, if possible.

For providing the main functionality, that is to consider the request made by a Data Seeker allowed, the component needs to access to:

- the information stored about the data seekers, including the one informed by them and the one calculated from previous experiences as a reputation score.
- the active policies of a specific data set in the DLT as part of the contracts.

This Engine does not provide a User interface to interact with it, considering it as an internal tool, part of the cloud platform and transparent to the users. This component exposes a function available for being called from the rest of the tools through an API. The implementation takes as input the IDs of the data sets requested and the attributes of the Data Seeker.

3.1.1 Technology Background

The Access Policy Engine is implemented in Java, using spring-boot framework and Swagger[5] for the specification and creation of the REST API.

[5] https://swagger.io/specification/

3.1.2 Component Backlog

3.1.2.1 Implemented Features

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_158	Access Policy Engine	DataVaults Cloud Platform	Identify the data involved in the request	APE accesses the policies associated to the selected data
US_159	Access Policy Engine	DataVaults Cloud Platform	Identify the attributes of a given Seeker	APE can use them for deciding on granting access or not.
US_160	Access Policy Engine	DataVaults Cloud Platform	Identify the information stored in the public ledger	APE retrieves any active sharing contracts, associated to the data asset
US_161	Access Policy Engine	DataVaults Cloud Platform	compare the Seeker's attributes with the access policies of the data	APE gives or denies access to the Seeker.
US_162	Access Policy Engine	Data Seeker	know why I was denied access, in case my request was denied	I can reconsider my profile attributes (ex. Submit documents to become a verified user)

3.1.2.2 Features planned for upcoming Releases

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_163	Access Policy Engine	DataVaults Cloud Platform	call the ABE mechanism for sharing data, if the access is granted (by execution of the smart contract), and the data are encrypted under the ABE scheme	the second access privacy and security layer is activated, to provide access to the user.
US_164	Access Policy Engine	DataVaults Cloud Platform	gather the information about the access request resolution process after a Seeker's access request	the Contract Composer can register the information in a contract regarding the transaction.
US_165	Access Policy Engine	DataVaults Cloud Platform	trigger the component for creating the contract for an access request resolution process and provide the related information	the responsible component for the creation of the contract registers the access authorisation/deny process.

These features will be substituted by new ones in case they are obsolete. The evolution of the project regarding the infrastructure may not include the flows for which these features were planned.

3.2 Risk Management Monitor and Dashboard

The Risk Management Monitor offers monitoring and evaluation of the risks related to privacy based on the sharing of datasets. It provides an overview of privacy risks and the privacy exposure to the platform administrator, and it also calculates the sharing exposure for each individual.

Risk Management Monitor is based on a model used to serialize all information that can describe the datasets shared in DataVaults and then performs the calculation of relevant risks. This privacy assessment is based on analysis of asset chains in order to estimate the impact that derives from the interconnections of the assets and datasets and provide a single estimation for the overall impact of a specific asset/vulnerability combination. The Risk Management Monitor also includes a graph and a list view, as depicted in the figure below.

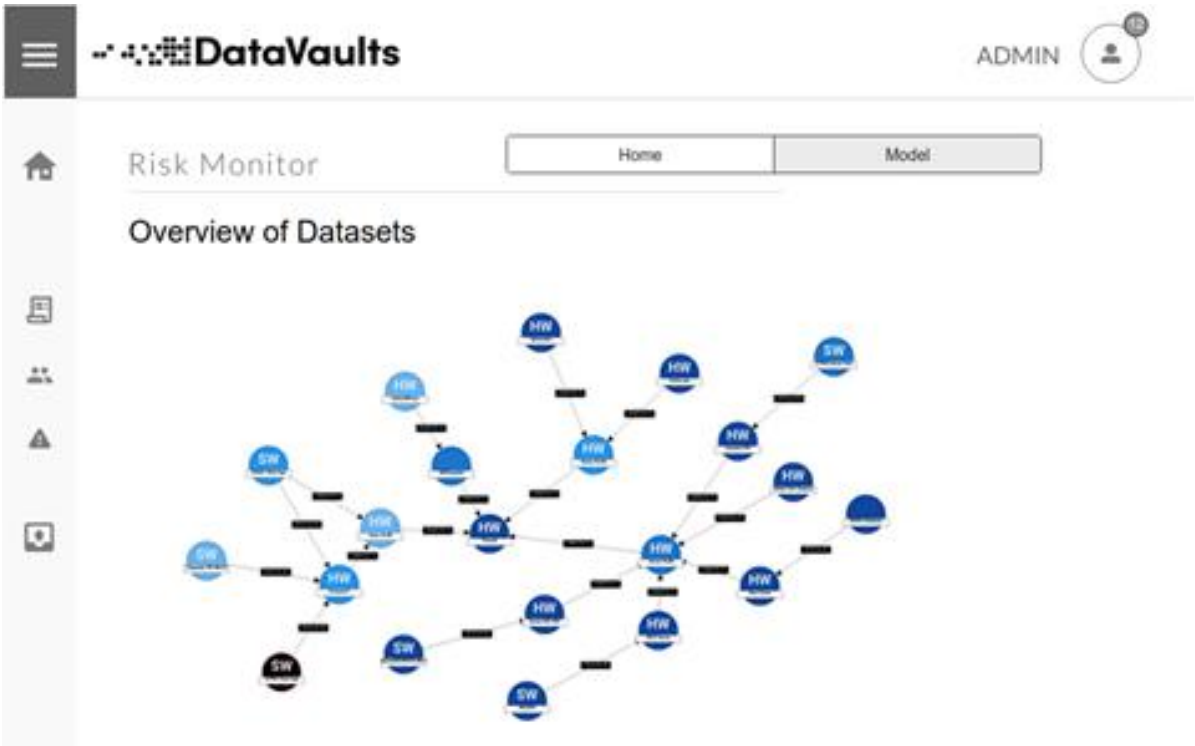

Figure 17: Mock-up of the privacy risk monitor

In addition, the Risk Management Dashboard provides to the user (through the Personal App) the calculated risk (that Risk Management Monitor provides) regarding the datasets that the user has added to the platform, and also a comprehensive view of current and previous privacy exposure degrees, as depicted in the figure below.

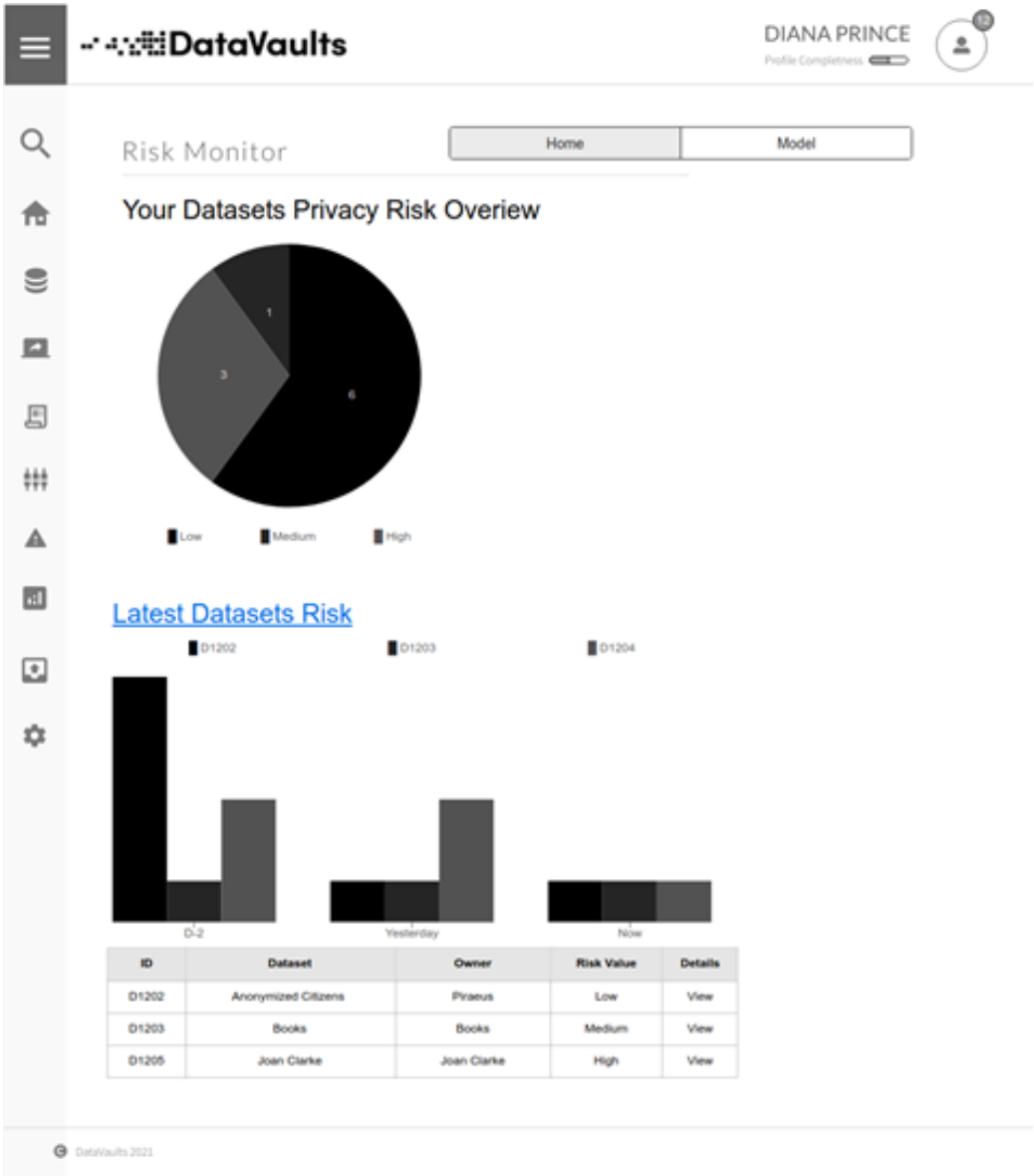

Figure 18: Mock-up of the privacy risk dashboard Technology Background

The Risk Management Monitor is a Java-based application that uses the meta-model of Privacy Assessment Tool (PAT)[6] of the *seCUre and pRivate hEalth data eXchange* (CUREX)[7] project. PAT is used as a backend tool for the modelling of the assets and also for the privacy quantification (of a dataset, an Individual and the overall platform). It is currently being extended to provide the following functionality:

- Allow calculation of the risk, based on the sharing aspects that DataVaults introduce
- Retrieve the datasets from the storage of Cloud Platform core backend
- Produce the appropriate UIs for the DataVaults Risk Management Monitor and the Risk Management Dashboard, based on the defined mockups and APIs

[6] https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/horizon-results-platform/26364
[7] https://curex-project.eu/

For easier integration to both the cloud platform and the personal app, Vue.js is used for the creation of the frontend part.

3.2.1 Component Backlog

3.2.1.1 Implemented Features

The features implemented at this point are supported at the backend level and also by early versions of the user interface. The creation of a proper user interface based on the mock-ups presented above is still under implementation.

ID #	Related Component	Related Epic	User Story		
			As a <Role>	I want to <Action>,	so that <Reason>
US_166	Risk Management Monitor	Risk Assessment	DataVaults Cloud Platform	assess the overall privacy exposure on the platform	I understand the current risk status
US_170	Risk Management Monitor	Risk Assessment	DataVaults Cloud Platform	assess the privacy exposure of certain user, based on the provided data	risk metrics are provided to the user
US_77	Privacy Metrics Dashboard, Risk Management Monitor	Risk Assessment	Data Provider	know the risk related to the data I want share to the Cloud Platform	I fix any privacy-related issues in the data sharing configuration.
US_79	Privacy Metrics Dashboard, Risk Management Monitor	Risk Assessment	Data Provider	know the privacy exposure of specific datasets that I have shared in the Cloud	I modify the sharing configuration or make the data completely unavailable for sharing.

3.2.1.2 Features planned for upcoming Releases

ID #	Related Component	Related Epic	User Story		
			As a <Role>	I want to <Action>,	so that <Reason>
US_167	Risk Management Monitor	Risk Assessment	DataVaults Cloud Platform	assess the privacy exposure of certain user, based on the provided data	risk metrics are provided to the administrator
US_168	Risk Management Monitor	Risk Assessment	DataVaults Cloud Platform	update the privacy exposure of certain user, based on the downloads of data by a Data Seeker	risk metrics are provided to the administrator
US_169	Risk Management Monitor	Risk Assessment	DataVaults Cloud Platform	assess the privacy exposure of certain user, based on the updates of the data	risk metrics are provided to the administrator
US_171	Risk Management Monitor	Risk Assessment	DataVaults Cloud Platform	update the privacy exposure of certain user, based on the downloads of data by a Data Seeker	risk metrics are provided to the user
US_172	Risk Management Monitor	Risk Assessment	DataVaults Cloud Platform	assess the privacy exposure of certain user, based on the updates of the data	risk metrics are provided to the user

US_78	Privacy Metrics Dashboard, Risk Management Monitor	Risk Assessment	Data Provider	view a graphical representation of the risk values related to my shared data	I modify the sharing configuration or make the data completely unavailable for sharing.
US_80	Privacy Metrics Dashboard, Risk Management Monitor	Risk Assessment	Data Provider	the privacy metrics of my data to be updated based on the downloads performed by a Data Seekers	I modify the sharing configuration or make the data completely unavailable for sharing.
US_81	Privacy Metrics Dashboard, Risk Management Monitor	Risk Assessment	Data Provider	know the overall privacy metrics of my user account	I modify the sharing configuration or make the data completely unavailable for sharing.

3.3 DATASTREAM AND CONTRACT COMPOSER

This component is responsible for the execution of data trading contracts whenever a Data Seeker is willing to acquire a dataset. The component is therefore used to request the instantiation of a data trading contract when an asset has an already fixed price, thus executing the contract as present in the ledger and recording a relevant transaction. This is an operation not visible to the user, as the contract is executed immediately without allowing any modification.

Additionally, this component handles data sharing requests when a Data Seeker selects to acquire either an already shared but not with a fixed price dataset, or new information from a Data Owner through a questionnaire builder, as shown in the figures below.

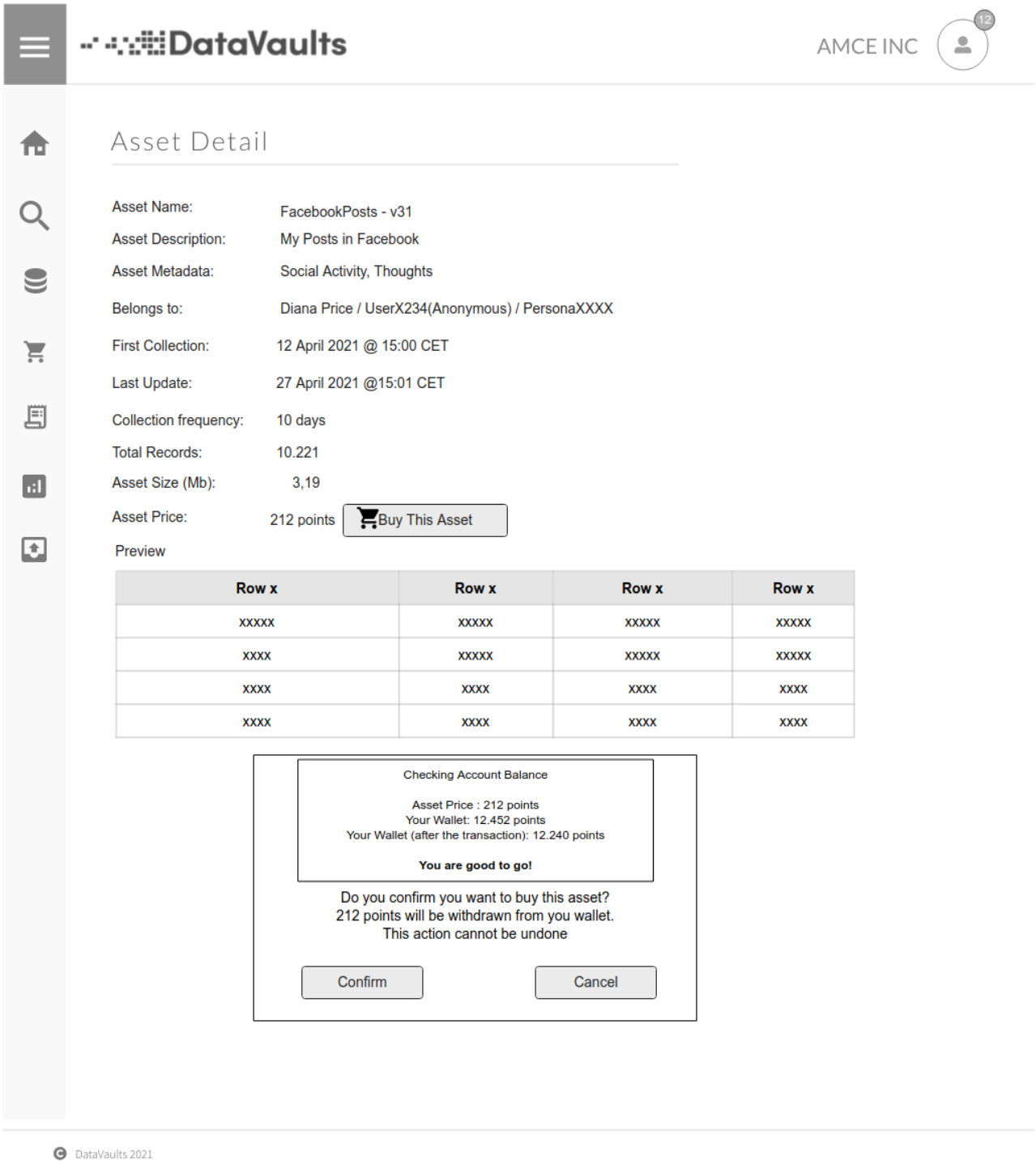

Figure 19: Mock-up of Acquiring a Dataset (from the Query Builder)

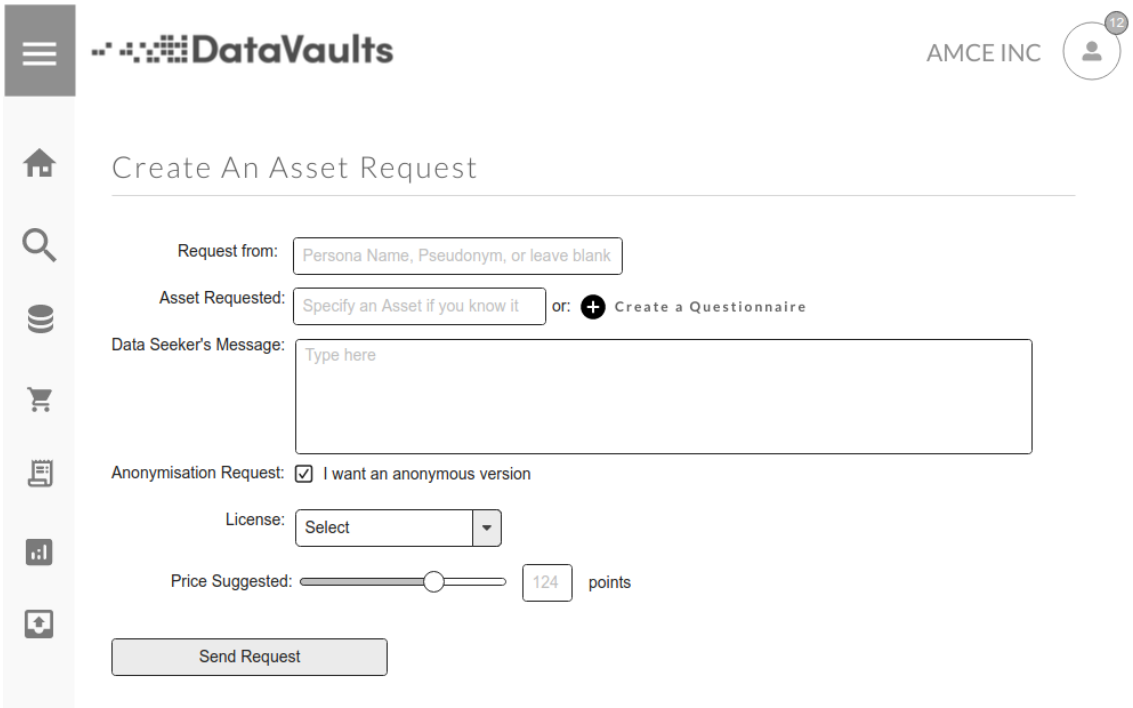

Figure 20: Mock-up of Requesting a Dataset

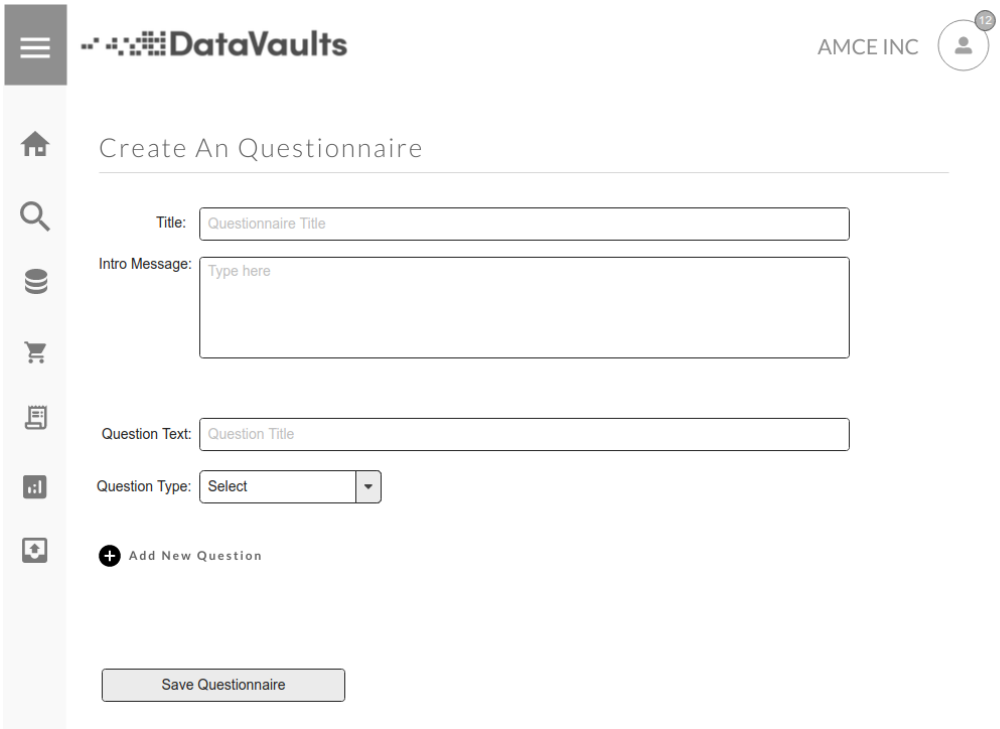

Figure 21: Mock-up of Building a Questionnaire Technology Background

The technology for implementing this component is VueJS2 and storage of the information that is necessary to build the questionnaire is done in the Postgres database which is located in the cloud-based infrastructure. The communication between this component and the Data Request Service resolver is done via RabbitMQ.

3.3.1 Component Backlog

3.3.1.1 Implemented Features

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>	so that <Reason>
US_186	DataStream & Contract Composer	Data Seeker	have a contract created every time I purchase data through DataVaults	the transaction and terms are recorded for logging and auditing purposes.
US_187	DataStream & Contract Composer	Data Seeker	have the contract stored in a secure manner	I am sure that it will not be tampered with.
US_188	DataStream & Contract Composer	Data Provider	have a contract created every time my data are purchased through DataVaults	the transaction and terms are recorded for logging and auditing purposes.
US_189	DataStream & Contract Composer	Data Provider	have the contract stored in a secure manner	I am sure that it will not be tampered with.
US_190	DataStream & Contract Composer	Data Seeker	be able to compose a draft contract for a request for data that are not yet available through DataVaults	I can make an offer to an Individual for data assets.
US_191	DataStream & Contract Composer / Notification System	Data Provider	receive a data sharing request with a predefined sharing configuration	I can quickly accept or decline the request
US_192	DataStream & Contract Composer	Data Provider	automatically share the requested asset in case I have accepted the request	I skip the sharing configuration step.
US_193	Notification System	Data Seeker	I want to receive a notification based on the outcome of a request	I can find out whether I possess the data or not

3.3.1.2 Features planned for upcoming Releases

All identified features have been delivered. If new user stories are identified during development and demonstrator validation (WP6), they will be included in subsequent deliverables.

3.4 Policy-compliant Blockchain Infrastructure and DLT Engine

This component is responsible for managing and maintaining the DataVaults **public and private ledgers** (as modelled in D2.2). It provides functionalities for the secure and accountable operational data monitoring, by recording the sharing activities between the Individuals, the DataVaults Cloud Platform and the Data Seekers via the use of smart contracts. The engine is responsible for enabling the necessary functionalities for on- and off-chain data and knowledge management services through the specification and provisioning of the appropriate interfaces related to secure, trusted and encrypted data trading. This is done by capturing all the data sharing behaviours that have been created by the Individuals, as well as the trusted consent management activities when data is exchanged/shared with requesting Data Seekers.

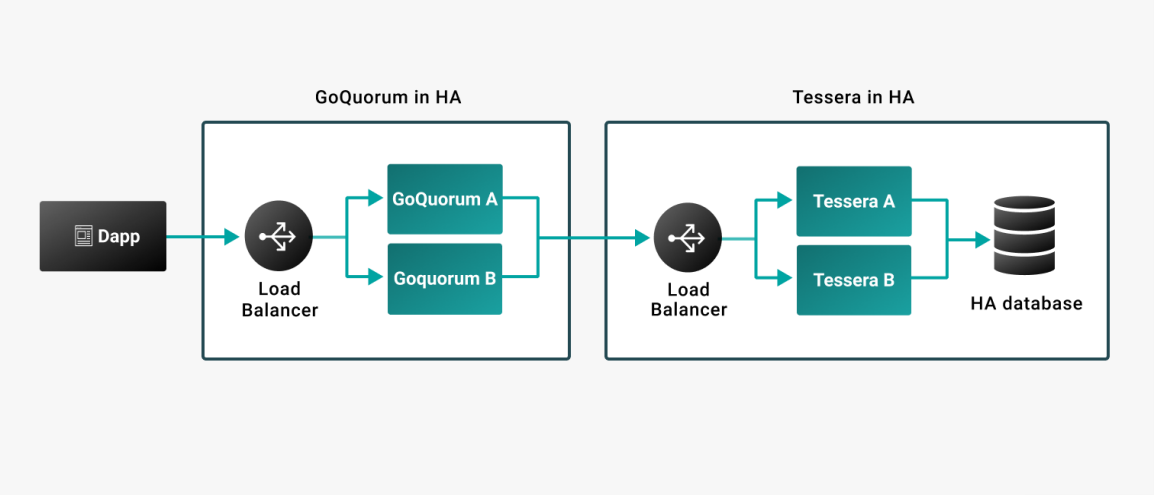

Figure 22: DataVaults Quorum DLT Conceptual Architecture

It is based on the use of the Quorum Technology following the conceptual architecture depicted in Figure 22. One of the main features provided by the DataVaults DLT Engine is the secure on- and off-chain management (creation, deployment and execution) of data trading smart contracts: capturing both the data sharing transactions (data uploaded by the Data Owners) as well as the data trading transactions initiated by the Data Seekers. Thus, there is a close interaction between the DLT Engine and other Data Vaults components responsible for providing the necessary information for the correct structuring of the smart contracts; i.e., DataStream and Contract Composer (Section 3.3), Sharing Configurator (Section 2.2), ABE Engine (Section 2.7), Access Policy Engine (Section 2.8).

Data sharing agreements and data trading transactions will be reflected on both the Individual's (permissioned) private ledger and mirrored to the (permissioned) public ledger so that all authorised participants in the decentralized data market can verify their correct execution. By permissioned, we mean that all recorded information flow will only be accessed by the authenticated entities via a membership access control layer that will be offered by the DLT Engine. The use of the ledgers is to ensure the data and event traceability across the entire data market and to be able to provide the required data security, user privacy and ledger security properties.

3.4.1 Technology Background

The DataVaults policy-compliant Blockchain infrastructure is based on the use of the open-source Quorum technology providing the below salient characteristics on the secure on- and off chain data management of safety-critical workloads:

- **DataVaults Permissioning:** Only registered and trusted users can join the DataVaults ecosystem and, thus, the underlying DLT infrastructures. This covers the entire spectrum of the already defined security requirements (Deliverable D2.2): From the certificate-based authentication, that is managed by the DataVaults Identity Provider, to the attestation-based authentication providing secure enrolment of users if and only if they are at a correct state (supported by the DataVaults Configuration Integrity Verification (CIV) mechanism – Section 2.1).
- **DataVaults Transaction/Contract Privacy:** DataVaults Quorum allows contracts to be deployed and transactions to be sent/shared only to a subset of participating users that abide to the defined access control policies;
- **DataVaults IBFT & RAFT Consensus:** Proof of authority-based consensus which provides immediate block finality, reduced time between blocks and high data integrity and fault tolerance. DataVaults creates blocks "on-demand," faster block times in the order of milliseconds instead of seconds and transaction finality (absence of forking).

The current implementation of the DataVaults Quorum-based DLT Engine (Figure 22) has extended the novel concept of "**Multi-Tenancy via Multiple Private States**". In a typical network, each participant (tenant) uses its own GoQuorum and Tessera node. Tessera can be configured to manage multiple key pairs which are owned by one tenant. This model is costly to run and does not scale as more tenants join the network, thus, limiting its applicability in the DataVaults ecosystem.

The "**Multi-Tenancy via Multiple Private States**" paradigm, followed in DataVaults, allows multiple tenants (Data Owners) to use the same GoQuorum node (Figure 23), with each user having their own private state(s). Users can perform all operations (create/read/write) on any contract in their private state and a single user can have access to multiple private states. Multi-tenancy allows for a similar user experience to a user running their own managed node. The public state remains available publicly to all tenants and private states are logically separated.

Figure 23: DataVaults Multi-Tenant via multiple Private States Quorum Node

The synchronization of multiple GoQuorum multi-tenant nodes was achieved through the integration of proxy servers acting as load balancers. Since a single tenant can have access to multiple private states that may be shared between different GoQuorum multi-tenant nodes (to achieve high scalability and parallelism in transaction management), all GoQuorum nodes need to have the same view of the entire Blockchain ecosystem, i.e. same state on all private and public transactions. Compounding this issue, DataVaults introduced appropriate load balancers for managing the load and synchronization of deployed GoQuorum nodes.

3.4.2 Component Backlog

3.4.2.1 Implemented Features

As can be seen from the table below, all features regarding the secure logging of all data trading transactions have been successfully implemented. In addition, the provision of the necessary APIs and mechanisms for supporting the execution of queries for retrieving data from the deployed contracts (e.g., dataset IDs of uploaded data, access policies for datasets with specific IDs, etc.) has also been finalized.

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_202	Private Ledger	Data provider	a record to be created each time my data are shared to the Cloud	the activity is securely logged.
US_203	Private Ledger	Data Provider	have transaction privacy	my personal information (configuration, account value) is hidden from other users.
US_207	Open Ledger	DataVaults Cloud Platform	restrict the access to the ledger	the functionalities are only provided to authorized Data Seekers.
US_208	Open Ledger	Data Seeker	have an open ledger account created when I register	I can make currency transactions to purchase data
US_209	Open Ledger	Data Provider	sell my asset only to verified data seekers	only trusted individuals receive my data.

3.4.2.2 Features planned for upcoming Releases

For the second release of the DataVaults DLT Engine the following features and enhancements are planned:

- Implementation of the DataVaults Brokerage Engine which is responsible for providing the DataVaults platform wallet for (temporarily) keeping the data trading value – committed from a Data Seeker – to be securely transferred to the respective Data Owner's personal wallet upon the successful execution of the transactions;
- Provision of APIs for the execution of more complex ledger reading queries on the (data trading) transactions stored on the distributed Ledgers: i.e., that are based on

information extracted from either multiple data structures in the same contract (e.g., "*What are the Dataset IDs procured by a specific Data Seeker based on a specific access policy?*") or multiple structures from different contracts (e.g., "*What is the total number of transactions executed by a specific Data Seeker and what is the overall resulting set of value tokens been exchanged?*")

Automation of the creation of additional Quorum Blockchain nodes (as part of the cloud-based DataVaults DLT Infrastructure) for supporting enhanced scalability – when the number of DataVaults registered users increases additional resources will also be deployed through appropriate containerizes Quorum Blockchain nodes.

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_204	DataVaults Private Brokerage Engine	Data Provider	have the agreed amount of currency transferred to my wallet	I receive the agreed compensation for my data.
US_205	DataVaults Private Brokerage Engine	Data Provider	use a privacy-preserving value transfer	my earnings are kept secret.
US_210	Open Ledger	Data Seeker	transact the agreed amount of currency from my wallet to DataVaults	I can purchase a data asset from a Provider.
US_211	Open Ledger	Data Seeker	display the value of my wallet account	I can evaluate the worth of the previous sharing activities

3.5 PERSONA GENERATOR

The DataVault's vision of a persona abandons the idea of traditional market research in favour of a modern data-driven approach. A **marketing persona** draws a picture of who your target audience is. Marketing personas are based on market **research via focus groups and interviews**, typically to represent the largest group you plan to target. Our data-driven approach is based on aggregation of data and the use of analytics (statistical and machine learning) to draw out insights and generalised characteristics that are representative of the data providers, from whom the aggregated data has been sourced. This process consists of taking a dataset of users and separating them into groups with similar demographic details. Insights from these groups can be found and used to generate a user persona. In this case, each persona would be a representative of a user group.

This process of persona generation is particularly of note to DataVaults as it would allow users to opt in to sharing their data as part of a group as opposed to a simple 1-to-1 anonymization process. By using the dataset to generate the insights incorporated into the persona, each user's identity would be concealed as no direct link can be made back to the dataset using the persona.

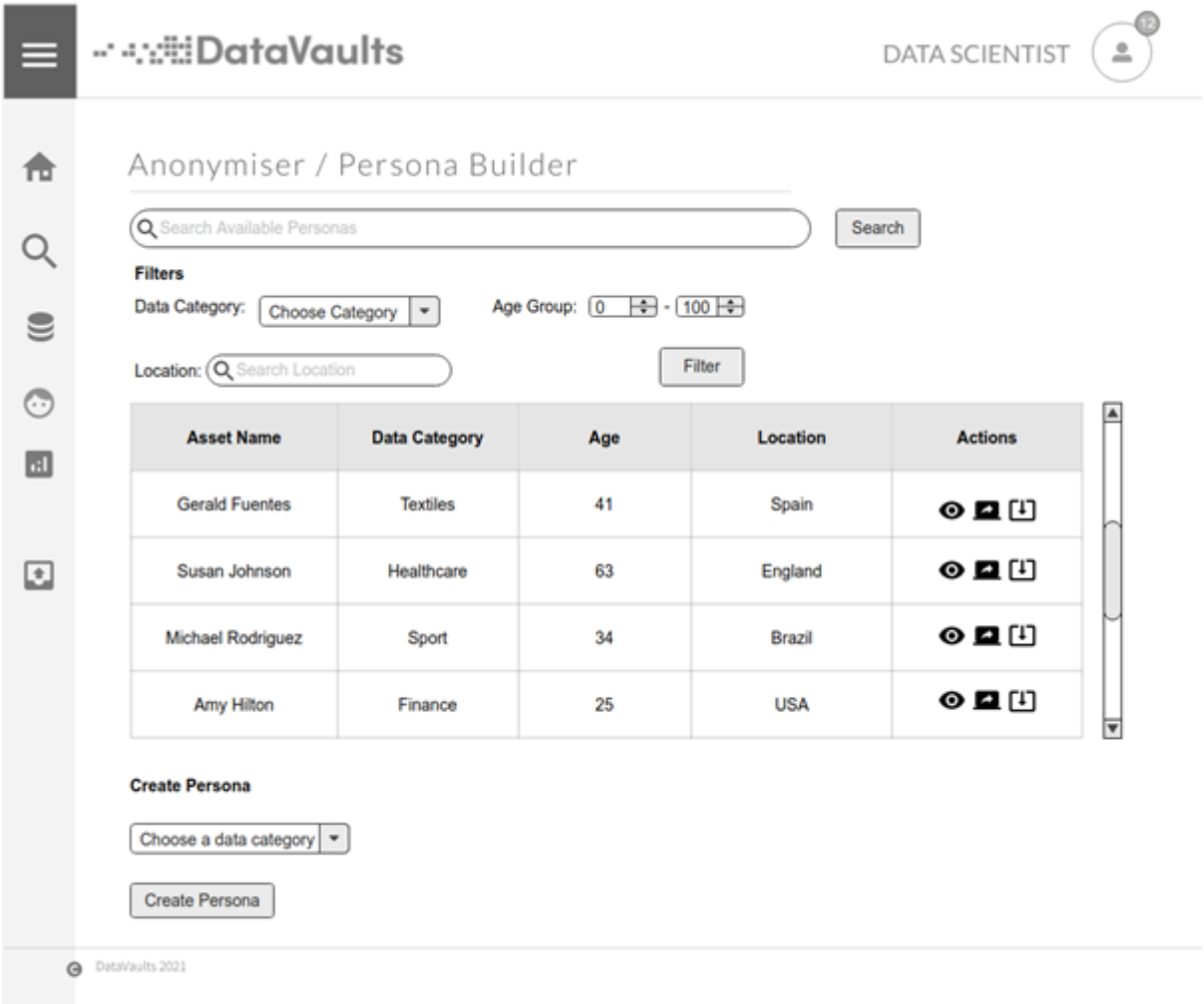

Figure 24: Persona Builder overview

3.5.1 Technology Background

The persona generator currently consists of three python scripts which carry out the primary function of the persona generator. We are using Dask[8] to fulfil our data processing needs such as encoding non-numerical values in the dataset, preparing the dataset for evaluation by the machine learning algorithm and extracting insights from the clustered data. Dask-ml is then used to apply a K-means algorithm to the dataset, which will separate the dataset into clusters, of which each cluster will then be represented by a persona. We then use PyPDF2[9], PyMUPDF[10], and Pillow[11] to produce a pdf containing a visual representation of our persona.

The workflow for the persona generator is as documented in the UML diagram in Figure 25.

[8] https://dask.org/
[9] https://pypi.org/project/PyPDF2/
[10] https://github.com/pymupdf/PyMuPDF
[11] https://python-pillow.org/

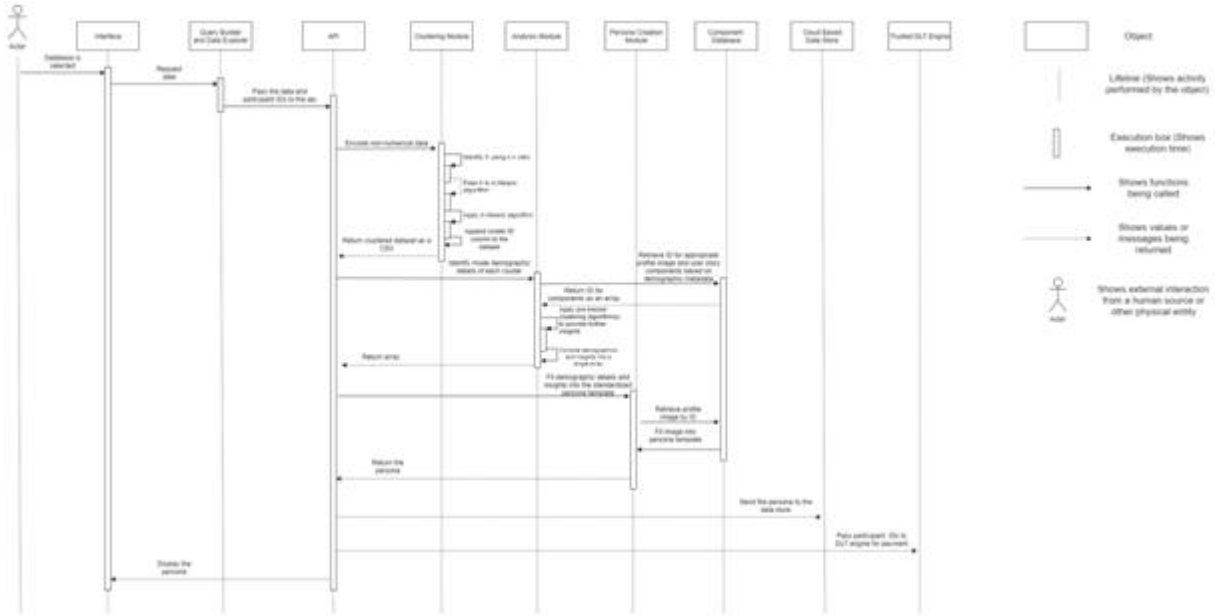

Figure 25: Workflow of Persona Generator

3.5.2 Component Backlog

3.5.2.1 Implemented Features

ID#	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_236	Persona Creation	Data Provider	share my data provided it is used for persona generation	the level of privacy is maintained
US_138	Insight_dask.py	Data Provider	be able to share insights as part of a group	my underlying personal data is not shared
US_237	Persona Creation	Data Provider	have my data assets that reside on the DataVaults Cloud Platform to be utilised in variants of persona creation	I remain completely anonymous
US_238	Persona Creation	Data Seeker	commission adjustments to a persona to a Data Analyst	he can adjust it accordingly to exactly match my specific needs
US_239	Persona Creation	Data Seeker	commission a	he can create a

			persona to a DataVaults Analyst	persona that exactly match my specific needs
US_240	Persona Creation	DataVaults Analyst	execute the Persona generation queries on the data stores on the platform	I can form the personas

3.5.2.2 Features planned for upcoming Releases

ID#	Related Component	User Story		
		As a <Role>	I want to <Action>	so that <Reason>
US_140	Persona Searching	Data Seeker	have access to personas generated	I can get insights on a particular group of people, demographic etc
US_241	Persona Searching	Data Seeker	search for personas and have access to a wide range of personas generated	I can be more economical than using the raw data.
US_242	Persona Searching	Data Seeker	see a list of all personas available in the DataVaults Cloud Platform	I can generate more specific understandings
US_243	Persona Web App	Data Seeker	select one of the personas and see all the information about this persona that is available for preview	I can explore the results and to be able to choose the most suitable
US_244	Persona Web App	Data Seeker	utilise personas	I can target those matching the persona
US_245	Persona Web App	Data Seeker	utilise personas	I can target those contributing to the persona in a way they remain unknown to me, ensuring their privacy

US_149	Additional Insights	Data Seeker	see how a persona has evolved over time	I identify any existing trends
US_246	Persona Creation	DataVaults Analyst	define different Personas based on queries on the data shared by Individuals (those shares using the Persona mode on)	I can have accurate personas based on actual questions being asked
US_247	Persona Creation	DataVaults Analyst	periodically update the Persona characteristics by re-running the persona generation queries,	new data values are considered and new members onboard the Persona and non-qualified members leave the Persona
US_248	Persona Creation	DataVaults Analyst	set a threshold for the minimum members necessary for a Persona to be constructed	we can ensure that data seekers are satisfied
US_249	Persona Creation	DataVaults Analyst	set a threshold for the maximum members necessary for a Persona to be constructed	we can ensure that data seekers are satisfied

4 CONCLUSIONS AND NEXT STEPS

This document describes the components from the DataVaults architecture assigned to WP3. The components provide the core functionalities to enable secure data sharing and access, as well as privacy and trust preservation.

Based on the release roadmap and development plan of the project, individual components have a varying degree of progress. This follows the prioritisation derived from the MVP to cover the requirements of the Alpha release of the project.

The current implementation state of the components will be incorporated into the verification and integration activities of WP5, while further development will take place in parallel. New features might be added to already defined components, or additional components could be introduced to the architecture. They may be the result of the demonstrator validation performed in WP6. Such alterations are expected and will be documented in the subsequent releases of this document.

This document will be updated to reflect the gradual implementation progress of the tasks. The next intermediate result will be delivered with D3.2, according to the project plan in M24.