



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

D4.3

Data Sharing, Value Generation and Intelligence Bundles - Version 3

Editor(s)	Sotiris Koussouris
Lead Beneficiary	Suite5 Data Intelligence Solutions
Status	Final
Version	1.0
Due Date	31/08/2022
Delivery Date	01/09/2022
Dissemination Level	PU



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	WP4 - Multitude Trusted Intelligence Bundles for Personal Data Insights Generation
Deliverable	D4.3 - Data Sharing, Value Generation and Intelligence Bundles - Version 3
Editor(s)	Suite5 – Sotiris Koussouris
Contributor(s)	FRAUNHOFER – Torben Jastrow, Simon Dutkowski ATOS – Tomás Pariente Lobo, Iván Martínez Rodríguez, Javier Villazán UBITECH – Giannis Ledakis, Konstantinos Oikonomou
Reviewer(s)	PIRAEUS – Michail Bourmpos, Erifili Baloti IFAG – Detlef Houdeau

Abstract	This deliverable is provided as an auxiliary document for the delivery of the code of the components of the WP4 activities, providing a high-level description of those
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

D4.3 is the third and final version of deliverable D4.1, whose objective is to present the status of the implemented service bundles of the project coming out of WP4.

The present deliverable is provided as an auxiliary document for the delivery of the code of the different components of DataVaults (which is the actual content of 4.3), as these are grouped under a deliverable of type OTHER.

The components that are presented in this deliverable are the following:

- The Data Fetcher and Transformation
- The Query Builder
- The Data Explorer
- The SSE Engine
- The Data Analytics Playground
- The Edge Analytics Engine

In this deliverable, the reader can find out more about the different components that have been developed under WP4 of the project till M32, and that are integrated into the final DataVaults platform by WP5.

It needs to be noted, that as WP4 ends in M32, a number of features whose completion has not been finalised and are still relevant for the delivery of the final DataVaults platform, will be provided under WP5. Those are clearly marked in the backlog of each component described below.

Table of Contents

- 1 Introduction 6
 - 1.1 Document structure 6
 - 1.2 Relation To Other WPs/Tasks 6
- 2 WP4 Components Descriptions 7
 - 2.1 Data Fetcher and Transformation 8
 - 2.1.1 Technology Background 8
 - 2.1.2 Component Backlog 8
 - 2.2 Query Builder..... 11
 - 2.2.1 Technology Background 12
 - 2.2.2 Component Backlog 12
 - 2.3 Data Explorer 15
 - 2.3.1 Technology Background 15
 - 2.3.2 Component Backlog 15
 - 2.4 SSE Engine..... 17
 - 2.4.1 Technology Background 17
 - 2.4.2 Component Backlog 17
 - 2.5 Data Analytics Playground 19
 - 2.5.1 Technology Background 21
 - 2.5.2 Component Backlog 21
 - 2.6 Edge Analytics..... 23
 - 2.6.1 Technology Background 24
 - 2.6.2 Component Backlog 24
- 3 Conclusions and Next Steps 26
- References..... 27

List of Figures

Figure 1: Query Builder - Search View Screenshot 11

Figure 2: Query Builder – Advanced Search View Screenshot..... 12

Figure 3: Data Explorer Screenshot..... 15

Figure 4: SEAS User Interface 19

Figure 5: SEAS Deployment User Interface 20

Figure 6: Miflow 20

Figure 7: Apache Superset..... 20

Figure 8: Edge (transaction) Analytics Screenshot..... 23

1 INTRODUCTION

The present deliverable is provided as an auxiliary document for the delivery of the code of the different components of DataVaults that are the direct outcome of WP4 (for M32 of the project), as these are grouped under a deliverable of type OTHER.

It is noted that for reasons of completeness, that in case the scope and the technology background of a component has not been changes, the texts describing those in this section and in the sections following are essentially the same as those presented in D4.1 [1]. In case there are changes, these are marked under a subsection in each description.

1.1 DOCUMENT STRUCTURE

This document starts with an introduction in Section 1, and Section 2 provides an overview for the different components that have been the focus of the work until M32 of the project. For each one, a short description is provided, including mock-up screens or real-life screenshots (depending on whether the component has a UI and based on the progress made in the UI design), and the technology stack for each implementation is briefly provided.

Moreover, for each component, the updated backlog of user stories coming out of WP5 is provided, identifying the user stories that have been completed so far in terms of development and indicating the ones which remain at the backlog and that will be considered for implementation in the upcoming releases.

Finally, section 3 of the documents provides a short concluding statement and describe the next steps.

1.2 RELATION TO OTHER WPS/TASKS

The D4.3 deliverable, as the final version of the WP4 series of deliverables, is closely related to the WP5 and WP3 deliverables. The different components and the user stories that drive their implementation are defined in WP5, and as such input is received from WP5.

In addition, the outcomes of D4.3 are provided as input to WP5, on two different levels:

- Firstly, the code of WP4 is provided as ready to integrate components into WP5 which will in turn handle the integration of those into the overall DataVaults solution alongside with their testing.
- Secondly, as the development of the different components and the tackling of the different user stories is following an agile approach in WP4, feedback is provided to WP5 which impacts the overall design and evolution of the architecture and might also impact the user stories.

Finally, as WP4 provides components that highly interact with those of WP3, a bidirectional communication channel between those WPs is established, and D4.3 is considering the outputs of D4.2 [2], while it also provides feedback to be used as input for the components that are developed under WP3 (and in this release are submitted as deliverable D3.3).

2 WP4 COMPONENTS DESCRIPTIONS

This section provides a high-level overview of the different components that are developed under WP4 of the project, according to the DoA and the latest developments of the DataVaults architecture, which mostly concern data management and analytics that are to be provided by the DataVaults Infrastructure.

In more detail, the components that are presented in the next subsections are the following:

- The Data Fetcher and Transformation, which is tasked with the collection of data from individuals, and as such is a component housed in the DataVaults Personal App.
- The Query Builder is a component operated by the Data Seekers that allows querying the Data Store of the cloud-based platform. It presents Data Seekers with a data catalogue and the contents of the data store (which they can access).
- The Data Explorer allows a Data Seeker to browse their own Data Spaces, where the assets they have already bought reside. This is a component, which is not depicted in the first version of the architecture but is going to be part of the second version.
- The Symmetric Searchable Encryption (SSE) Engine, which is tasked with performing the necessary activities for encrypting and decrypting the various data based on SSE scheme.
- The Data Analytics Playground that is used by Data Seeker for designing and running analytics jobs based on the datasets they have acquired.
- The Edge Analytics Engine, which is a lightweight analytics implementation, offered to Data Owners to get insights on the data that reside in their Personal App.

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

<https://www.gitlab.com/DataVaults>

2.1 DATA FETCHER AND TRANSFORMATION

Component's Concept Update from D4.1/D4.2:

Component's Description: The Data Fetcher & Transformer component utilizes a modular micro-service architecture. The fetcher module can periodically check an Application Programming Interface (API), which is selected by a user (out of a list of available APIs in the Personal App which can connect to external sources) to find data and collect them. At this stage, only a small subset of predefined APIs is supported. The data then is transformed into the DataVaults metadata data model and exported into a MongoDB¹ for further usage.

In its current form, it is possible to configure the schedule for a preconfigured data source and to start and stop the collection schedule. The Data Fetcher & Transformer does not have a user interface, as all configuration is done via the DataVaults Personal App interface. The configurations for the Data Fetcher and Transformer regarding the user management reside on the backend on a special multiuser scheme.

2.1.1 Technology Background

Technology Background Update from D4.1/D4.2:

Technology Description: The Data Fetcher & Transformer is programmed in Java and Kotlin code. It uses Eclipse Vert.X² as web framework. The resulting micro-services are based on *piveau consus*³.

2.1.2 Component Backlog

This section provides the full set of features that belong to the backlog, splitting those in the ones already implemented, and the ones that are pending implementation.

2.1.2.1 Implemented Features (delivered in the v0.50 Release)

The table below provides the list of the features that have been delivered as part of the v0.50 release, as well as the list of features that have been part of the Alpha and the Beta release. The latter have been subject to optimisations and bug fixing, following the continuous integration and agile development approach followed by the project. New user stories that were not part of the initial backlog provided in D4.1 are marked with an asterisk (*) next to their ID. It is noted that user stories that are deemed as obsolete have been removed from the backlog.

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>

¹ <https://www.mongodb.com/>

² <https://vertx.io/>

³ <https://www.piveau.de/en/>

Features provided as part of the v0.50 Release				
No new features released - The main development for this release are the implementation of new data sources and improvements to existing ones.				
Features provided as part of the Beta Release				
US_33	Data Fetcher & Transformation	Data Provider	manually select the specific data that will be collected from the connected data source	I have control on which data assets are collected by my DataVaults Personal app and exclude any data that I don't want to be transferred from the data source.
US_42	Data Fetcher & Transformation	DataVaults Personal App	link different datasets from the collection tasks of one account	I know that these refer to the same Individual
Features Implemented by the Alpha Release (and optimised if needed in Beta Release)				
US_34	Data Fetcher & Transformation	Data Provider	execute a data collection configuration	I can collect data from the connected data source.
US_36	Data Fetcher & Transformation	Data Provider	add a collection schedule to a collection configuration	I can configure a recurring collection process for the specific data source (e.g., once a day, once a week etc.)
US_38	Data Fetcher & Transformation	Data Provider	pause a collection schedule	I can stop my DataVaults personal App from collecting data from a specific data source for a time period and if I change my mind later, restart data collection using the same configuration
US_40	Data Fetcher & Transformation	Data Provider	remove a collection schedule from a saved configuration	the collection no longer happens automatically and recursively..
US_45	Data Fetcher & Transformation	DataVaults Personal App	I want to transform the fetched data into the DataVaults data model	I can ensure compliance with the available DataVaults schemas as well as the necessary data quality levels to achieve maximum usefulness and usability.

2.1.2.2 Features to be released in the v1.00 version of the platform

The table below the features that remain in the backlog and are scheduled to be delivered as part of the v1.00 version of the platform, delivered by WP5.

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>

US_35	Data Fetcher & Transformation	Data Provider	save a data collection configuration	I can reuse it in the future.
US_37	Data Fetcher & Transformation	Data Provider	configure how data from the intervals between recurring collections will be handled	I can configure the "density" of the data that are collected in my DataVaults personal App.
US_39	Data Fetcher & Transformation	Data Provider	edit a saved data collection configuration	I can change the selected data, collection schedule etc.
US_41	Data Fetcher & Transformation	Data Provider	delete a data collection configuration	I completely discard a collection configuration I no longer need
US_43	Data Fetcher & Transformation	DataVaults Personal App	semantically enrich the collected data	I can retrieve similar data from different sources with different representations.
US_44	Data Fetcher & Transformation	DataVaults Personal App	do a Quality Check over the fetched data	I know, how to transform the data into the DataVaults data model.

2.2 QUERY BUILDER

Component's Concept Update from D4.1/D4.2:

Component's Description: The Query Builder is the main facility used by Data Seekers for searching within the cloud-based platform to retrieve datasets, which are shared by the different data owners. In the backend, the Query Builder uses a Triple Store for Linked Data to store, retrieve and search information about the data that is retrieved via the frontend, while the core of the dataset is stored in a non-relational database.

This approach enables a hybrid solution that allows semantic search and interoperability of data at the metadata level, providing functionalities such as those of a data catalogue. At the same time the core datasets can be indexed (if required) and served in a faster and more efficient manner to the requested component, directly from the non-relational data store.

For this purpose, apart from the usual APIs to communicate with the non-relational data store, an additional micro-service provides a RESTful API that translates frontend requests into SPARQL queries.

The frontend interface of the Query builder is based on VueJS to match it with the UX of the rest of the DataVaults cloud-based platform, and various operations that have to do with calling APIs from other components are handled by the frontend. In this respect, data retrieval is performed by the backend APIs and in turn communicating their results with the APIs of the Access Policy Engine, as access to the queried data is resolved by the Access Policy Engine, by utilizing as input information from the ledger. Moreover, this component interacts also with the SSE Engine, to allow searching over encrypted data in the later releases of the platform.

Some screenshots of the component, as delivered in the v0.50 release, follow.

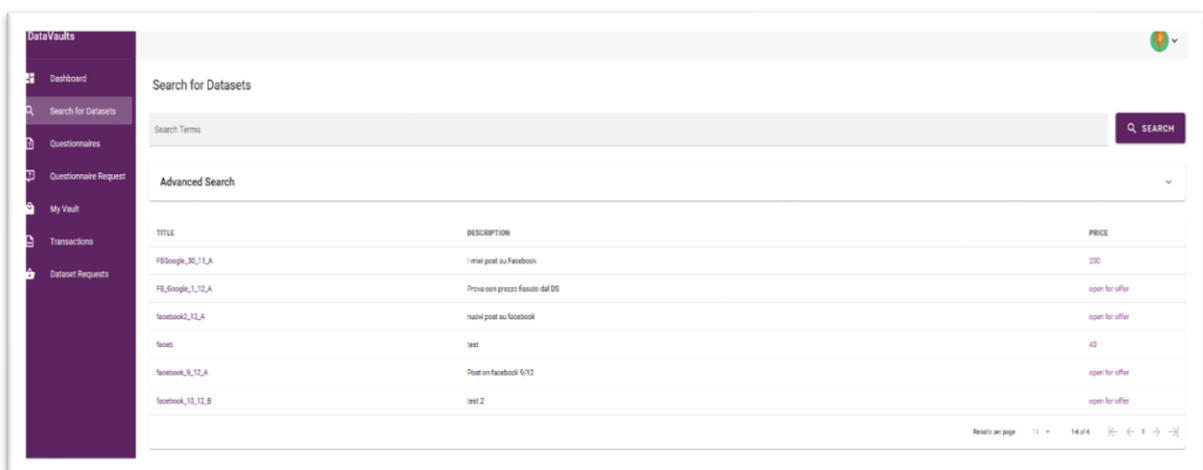


Figure 1: Query Builder - Search View Screenshot

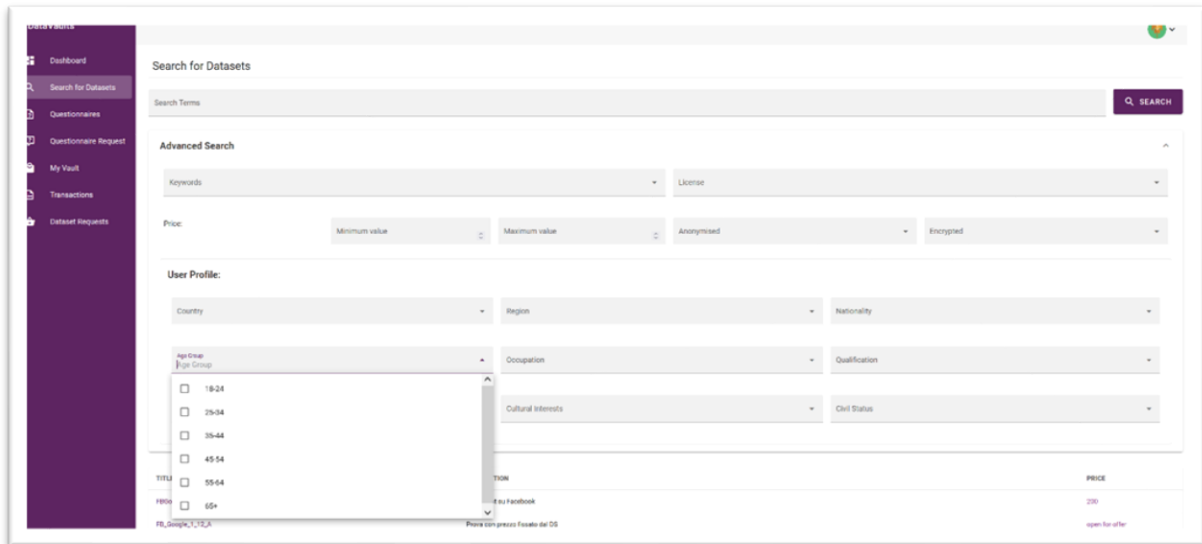


Figure 2: Query Builder – Advanced Search View Screenshot

2.2.1 Technology Background

Technology Background Update from D4.1/D4.2:

Technology Description: The backend of the Query Builder is written in Java and Kotlin and uses the Eclipse Vert.X web framework. For data storage, an OpenLink Virtuoso⁴ server is used. The Backend is based on *piveau hub*. The frontend of the Query builder is provided using the VueJS2⁵ framework.

2.2.2 Component Backlog

2.2.2.1 Implemented Features (delivered in the v0.50 Release)

The table below provides the list of the features that have been delivered as part of the v0.50 release, as well as the list of features that have been part of the Alpha and the Beta release. The latter have been subject to optimisations and bug fixing, following the continuous integration and agile development approach followed by the project. New user stories that were not part of the initial backlog provided in D4.1 are marked with an asterisk (*) next to their ID. It is noted that user stories that are deemed as obsolete have been removed from the backlog.

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
Features provided as part of the v0.50 Release				

⁴ <http://virtuoso.openlinksw.com>

⁵ <https://vuejs.org/>

US_218	Query Builder	Data Seeker	search the data assets available in the DataVaults Cloud based on their content, using keywords	I can easily find data assets that contain information that could be useful for me.
US_219	Query Builder	Data Seeker	search the data assets available in the DataVaults Cloud Platform using complex queries, i.e., by defining more than one criterion in my search	I can find data assets that match to various aspects.
Features provided as part of the Beta Release				
US_214	Query Builder	Data Seeker	filter the enlisted data assets by choosing from the available values in various categories (such as domain - e.g., health -, format -e.g., json-, anonymisation level - e.g., none, digital twin, persona-, language - e.g., GR, DE,- etc.)	I can limit the list of enlisted data assets to those that match my needs.
Features Implemented by the Alpha Release (and optimised if needed in Beta Release)				
US_213	Query Builder	Data Seeker	purchase/acquire one of the enlisted data assets	I get access to the data.
US_215	Query Builder	Data Seeker	search the data assets available in the DataVaults Cloud based on their name, using keywords or the exact title	I can easily find data assets that I know their names or their name indicates they could be useful for me.
US_216	Query Builder	Data Seeker	search the data assets available in the DataVaults Cloud based on metadata (e.g., description, author, domain, tags) using keywords	I can easily find data assets that their metadata match my needs.
US_217	Query Builder	Data Seeker	search the data assets available in the DataVaults Cloud based on metadata such as age group, price, etc. using ranges	I can easily explore data assets that their metadata match my needs, even when I do not want to provide an exact value, but I am interested in a group.
US_220	Query Builder	Data Seeker	see a list of all the data assets matching to my search criteria	I can explore the results of my search.
US_221	Query Builder	Data Seeker	select one of the enlisted data assets and see all the information about this data asset that is available for preview	I can see if this data asset contains the information I need.

2.2.2.2 Features to be released in the v1.00 version of the platform

The table below the features that remain in the backlog and are scheduled to be delivered as part of the v1.00 version of the platform, delivered by WP5.

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_222	Query Builder	Data Seeker	save queries	I can use them again in the future.
US_223	Query Builder	Data Seeker	load and edit saved queries	I can add or remove any search criteria that I do not need for my new search.
US_224	Query Builder	Data Seeker	delete saved queries	I can remove a query I no longer need

2.3 DATA EXPLORER

Component’s Concept Update from D4.1/D4.2:

Component’s Description: Initially part of the larger component called “Query Builder and Data Explorer”, this component is the one allowing Data Seekers to browse into their own Data Space and retrieve, delete or forward for analysis the data assets they have already bought. Figure 3 show a screenshot.

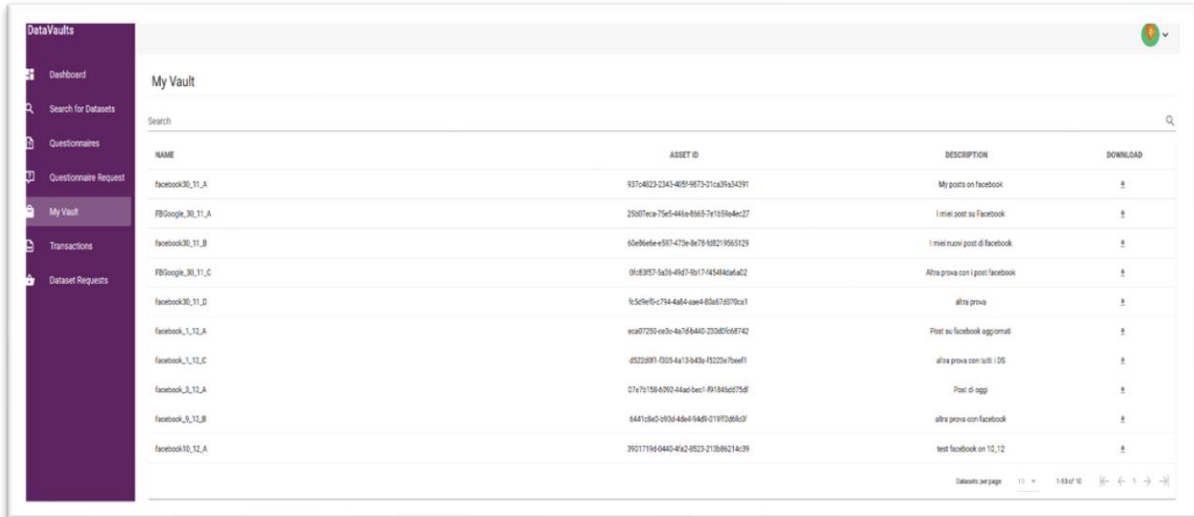


Figure 3: Data Explorer Screenshot

2.3.1 Technology Background

Technology Background Update from D4.1/D4.2: The main technology background for this component has been shifted to Java from Python, to allow for better interoperation with other components of the backend which are also Java applications

Technology Description: The Data Explorer is written in Java, using VueJS2 in the frontend.

2.3.2 Component Backlog

2.3.2.1 Implemented Features (delivered in the v0.50 Release)

The table below provides the list of the features that have been delivered as part of the v0.50 release, as well as the list of features that have been part of the Alpha and the Beta release. The latter have been subject to optimisations and bug fixing, following the continuous integration and agile development approach followed by the project. New user stories that were not part of the initial backlog provided in D4.1 are marked with an asterisk (*) next to their ID. It is noted that user stories that are deemed as obsolete have been removed from the backlog.

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
Features provided as part of the v0.50 Release				
US_233	Data Explorer	Data Seeker	upload my own data through an API	I can use them for analytics.
Features provided as part of the Beta Release				
US_232	Data Explorer	Data Seeker	upload my own data as files	I can use them for analytics.
US_234	Data Explorer	Data Seeker	search for specific data assets using keywords	I can easily explore my data assets.
Features Implemented by the Alpha Release (and optimised if needed in Beta Release)				
US_230	Data Explorer	Data Seeker	browse through the assets I have already purchased/stored in my storage container	I can use it in the analytics containers
US_231	Data Explorer	Data Seeker	select from available options (e.g., 'Group by source', 'Group by domain', etc.) to view my data	I can group my data.

2.3.2.2 Features to be released in the v1.00 version of the platform

The table below the features that remain in the backlog and are scheduled to be delivered as part of the v1.00 version of the platform, delivered by WP5.

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_235	Data Explorer	Data Seeker	filter data assets based on metadata values	I can easily explore my data assets.

2.4 SSE ENGINE

Component’s Concept Update from D4.1/D4.2: None

Component’s Description: The SSE Engine of DataVaults aims to encrypt metadata of files and then allow users to search on top of those encrypted metadata, adopting the approaches used in the ASCLEPIOS project, where metadata and data content from selected files are encrypted using a symmetric key and the cloud-based service can perform queries over the store ciphertext, without knowing the keys to decrypt them. SSE at the first stage will be applied at the metadata level, as this is considered more efficient from a resource point of view, and keys of SSE will be protected through the ABE Engine, so that only Data Seekers with the appropriate attributes will be in a position to search over the encrypted metadata. As such, integration with the ABE Engine is planned, and will be delivered in the next stages of the project.

During this period, work has focused on analysing the possible implementation options and building the data flow and the appropriate infrastructure upon which the SSE Engine will be developed, and development of the key features has started. A first version of this component is expected to be released in the Beta release of the platform.

The SSE Engine does not have a visual interface and is fully a back-end service used by the Query Builder, operating without the Data Owner or Seeker needing to configure anything, as keys are automatically issued and used, once a Data Owner chooses to provide his/her data in an encrypted manner to DataVaults.

2.4.1 Technology Background

Technology Background Update from D4.1/D4.2: None

Technology Description: The technology used in the SSE Engine is based on Python code, reusing libraries developed in the frame of the ASCLEPIOS H2020 project

2.4.2 Component Backlog

2.4.2.1 *Implemented Features (delivered in the v0.50 Release)*

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_03	Encryption/Decryption Engine	Data Provider	be sure that the encryption keys used to secure my data will not be made available to anyone else when I decide to share my data through the Cloud	my data that are protected with cryptographic techniques are not at risk.

US_107	DV Cloud Platform Backbone. Encryption / Decryption	Data Provider	have selected data assets that reside on the DataVaults Cloud Platform to be stored encrypted	only I and Data Seekers with the appropriate rights can get access to them.
--------	--	---------------	---	---

2.4.2.2 Features to be released in the v1.00 version of the platform

The table below the features that remain in the backlog and are scheduled to be delivered as part of the v1.00 version of the platform, delivered by WP5.

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_108	DV Cloud Platform Backbone. Encryption / Decryption	Data Provider	be sure that the decryption keys provided to Data Seekers for getting access to my data assets, are configured so that they enforce the sharing terms in place (e.g., expire after a period, are only for one use etc.)	the sharing of my data assets complies to the sharing terms I have defined.
US_109	DV Cloud Platform Backbone. Encryption / Decryption	Data Seeker	be able to search over the encrypted data assets available over the DataVaults Cloud	I can find any data assets of interest.

2.5 DATA ANALYTICS PLAYGROUND

Component's Concept Update from D4.1/D4.2: None

This deliverable, as is named, describes the possible updates of the different components of the platform. Atos has minor updates for its component's concept in this period, so the below text has no huge difference from what is written in Deliverable 4.1 "Data Sharing, Value Generation and Intelligence Bundles - Version 1" and Deliverable 4.2 "Data Sharing, Value Generation and Intelligence Bundles - Version 2".

Component's Description: The SEAS is a module inside DataVaults platform that offers the possibility of running different algorithms over the shared data in a secure way.

The SEAS is a module developed for being used by the Data Seekers however Data Providers could utilize it to gain insights into their own data. The module is divided into two subcomponents. The first subcomponent "Service Analytics Host" is in charge of defining the features of the setup and configuration of the Playground and it has a UI to manage and define these characteristics, as shown in Figure 4. The user is able to deploy the playground on a defined server or local, for example on their own laptop. Figure 5 shows how this process could be done. The Data Seeker has the Playground and Visualization modules in an isolated environment to create their own models or algorithms or to select the ones provided by the platform

The second subcomponent "Playground & Visualization Host" allows the Data seeker to execute and track the Machine Learning (ML) models (Figure 6). The tracking capability allows the user to decide which one of these will be stored in the Cloud, sent to the Visualization module (Figure 7), or both.

For this version, one of the objectives is aimed to give a better user experience. A Single Sign On (SSO) has been implemented inside the V0.5 to achieve this target. This SSO allows the user to execute all the functionalities of DataVaults using one unique User and Password, which facilitates the navigation along the DataVaults Platform.



Figure 4: SEAS User Interface

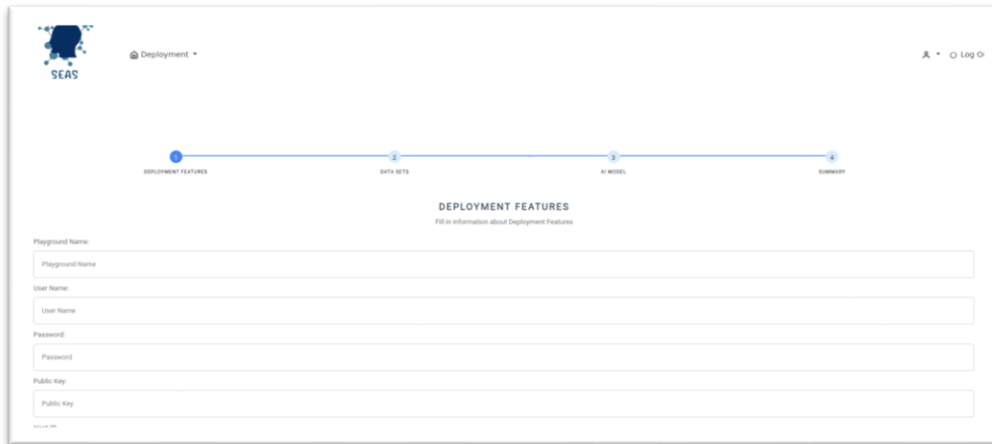


Figure 5: SEAS Deployment User Interface

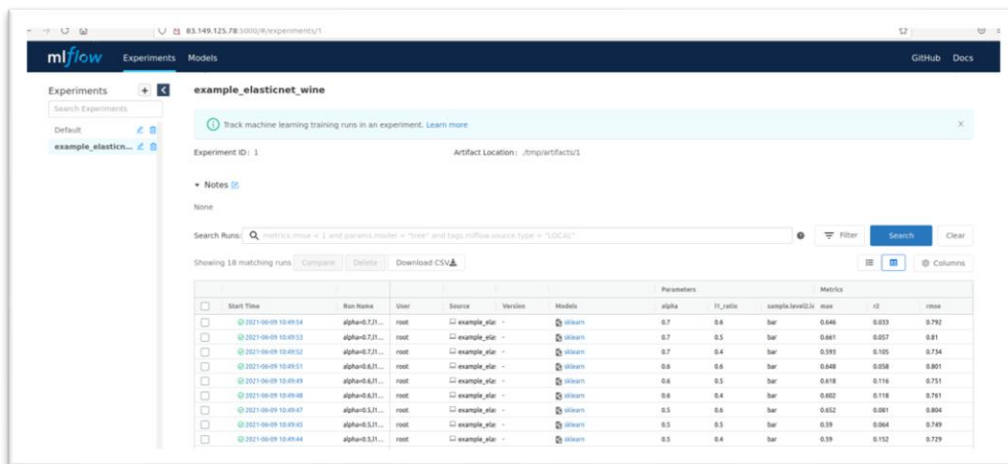


Figure 6: MLflow

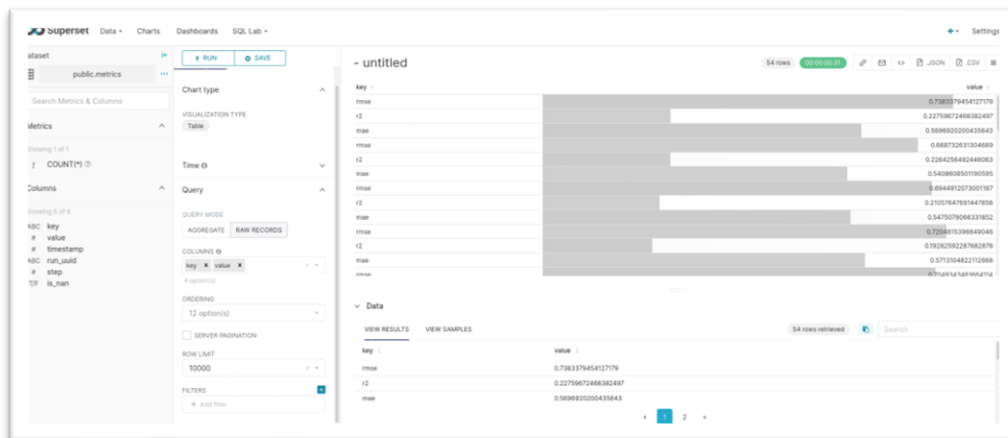


Figure 7: Apache Superset

2.5.1 Technology Background

Technology Background Update from D4.1/D4.2:

This deliverable, as is named, describes the possible updates of the different components of the platform. Atos has minor updates for its technology background in this period, so the below text has no huge difference from what is written in Deliverable 4.1 "Data Sharing, Value Generation and Intelligence Bundles - Version 1" and Deliverable 4.2 "Data Sharing, Value Generation and Intelligence Bundles - Version 2".

Technology Description: The "Service Analytics Host" uses technologies such as Play for the User Interface, Java for the backend setup and Docker for the deployment. Specific Java libraries have been used to create the integration between the Keycloak deployed in the DataVaults platform and the Graphic Unit Interface (GUI) of the SEAS. Ansible allow us to execute the different tasks associated with the deployment on various platforms (e.g. local or on a defined server).

The "Playground & Visualization Host" is based on Mlflow⁶ to execute and track the Machine Learning models, PostgreSQL⁷ to storage the results of the models and Apache Superset to visualize the results

2.5.2 Component Backlog

2.5.2.1 Implemented Features (delivered in the v0.50 Release)

The table below provides the list of the features that have been delivered as part of the v0.50 release, as well as the list of features that have been part of the Alpha and the Beta release. The latter have been subject to optimisations and bug fixing, following the continuous integration and agile development approach followed by the project. New user stories that were not part of the initial backlog provided in D4.1 are marked with an asterisk (*) next to their ID. It is noted that user stories that are deemed as obsolete have been removed from the backlog.

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
Features provided as part of the v0.50 Release				
US_183	Secure Analytics Playground	Data Seeker	download the results of an analysis to my local device	I can use them offline
US_185	Secure Analytics Playground	Data Seeker	download a visualisation to my local device	I can use it offline
US_250 *	Secure Analytics Playground	Data Seeker	fetch into the SEAS engine data assets that reside in my Vault (of the cloud platform)	I can perform analytics on those data

⁶ <https://mlflow.org/>

⁷ <https://www.postgresql.org/>

Features provided as part of the Beta Release				
US_176	Secure Analytics Playground	Data Seeker	run an analytics task over my requested data	I can get my needed information from DataVaults
US_179	Secure Analytics Playground	Data Seeker	be able to run my own algorithms on the cloud	I can easily use specialized algorithms over the gathered data
US_181	Secure Analytics Playground	Data Provider	see graphics and visualizations of the algorithms results	I can gain some insights about my data
US_184	Secure Analytics Playground	Data Seeker	save a visualisation in my storage container	I can use it in the future
Features Implemented by the Alpha Release (and optimised if needed in Beta Release)				
US_173	Secure Analytics Playground	Data Seeker	be able to configure an analytics task	I can perform analytics and gain insights from my data.
US_174	Secure Analytics Playground	Data Seeker	select the data on which I want to run an analysis	I can generate intelligence out of the data purchased
US_175	Secure Analytics Playground	Data Seeker	configure and execute a visualisation task	I can gain visual insights from my data.
US_177	Secure Analytics Playground	Data Seeker	run aggregation/filtering algorithms on the cloud	the amount of data I need to download is reduced
US_178	Secure Analytics Playground	Data Seeker	see graphics and visualizations of the analytics task	I avoid running visualisations in-house

2.5.2.2 Features to be released in the v1.00 version of the platform

The table below the features that remain in the backlog and are scheduled to be delivered as part of the v1.00 version of the platform, delivered by WP5.

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
US_180	Secure Analytics Playground	Data Provider	run algorithms over my own data and compare it to other "open" data in the platform	I can gain some insights about my data

US_182	Secure Analytics Playground	Data Seeker	download the results of an analysis to my storage container	I can reuse them in the future
--------	-----------------------------	-------------	---	--------------------------------

2.6 EDGE ANALYTICS

Component’s Concept Update from D4.1/D4.2: Based on inter-consortium discussions and after consultation with end users and demonstrators, a pivot has been made to the initial idea of having incoming data analysed at the Personal App level. As such, the consortium decided to steer the Edge Analytics component towards providing baseline analytics on the type of data an individual has collected, and the activities he has performed over DataVaults. In this direction, the component at this point is transparent to the user, and the analyses provided to each user are not configurable (by the user), but follow the same, general algorithms for all users with regards to their data and the transactions of the users (e.g. of the data owners)

Component’s Description: The Edge Analytics component is offered as a component of the DataVaults Personal App tasked to provide Data Owners with ready-made visualisations concerning their data as well as some analytics regarding their activity within DataVaults (see Figure 8).

This is performed by identifying specific data sources for which predefined analytics will be set up, so that the users can view relevant metrics without asking them to interact with the component. At the backend, this component includes an analytics library and engine, allowing the DataVaults developers to also set up also other types of analytics workflows and execute these flows on their side, to generate intelligence which can be shown to the data owners with the introduction of new visualisations.

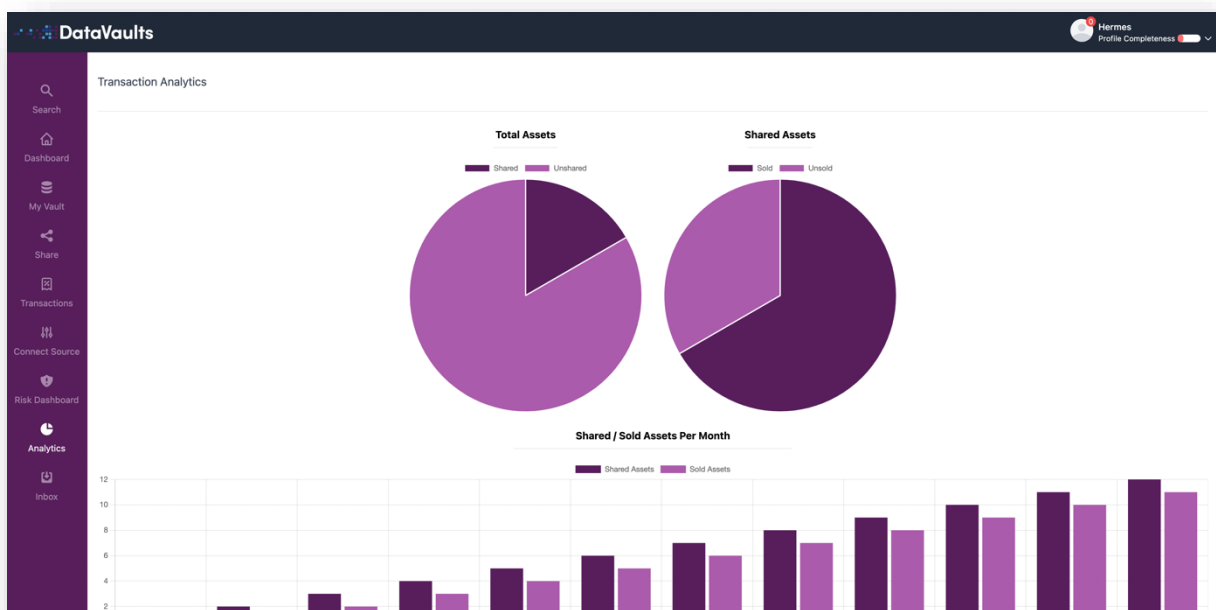


Figure 8: Edge (transaction) Analytics Screenshot

2.6.1 Technology Background

Technology Background Update from D4.1/D4.2: Due to the nature of analytics to be executed, a shift to the technology stack of Pandas and scikit-learn has been decided, as these are lightweight and can easily fit to the concept of the analytics to run at the side of the data owners.

Technology Description: The Edge Analytics component shall be based on the Pandas⁸ library and scikit-learn⁹. The frontend of the component is implemented using the VUEJS3 framework

2.6.2 Component Backlog

2.6.2.1 Implemented Features (delivered in the v0.50 Release)

The table below provides the list of the features that have been delivered as part of the v0.50 release, as well as the list of features that have been part of the Alpha and the Beta release. The latter have been subject to optimisations and bug fixing, following the continuous integration and agile development approach followed by the project. New user stories that were not part of the initial backlog provided in D4.1 are marked with an asterisk (*) next to their ID. It is noted that user stories that are deemed as obsolete have been removed from the backlog.

ID #	Related Component	User Story		
		As a <Role>	I want to <Action>,	so that <Reason>
Features provided as part of the v0.50 Release				
US_87	Edge Analytics Engine	Data Provider	view on my device the visualisation results of the defined visualisation task	I can easily understand my activity and progress in relation to factors such as time.
US_252*	Edge Analytics Engine	Data Provider	view statistics and analytics of my activities over DataVaults	I can understand how I use the platform
US_253*	Edge Analytics Engine	Data Provider	change the time granularity of the view in certain graphs	I can get insights broken down by month/week/etc
Features provided as part of the Beta Release				

⁸ <https://pandas.pydata.org/>

⁹ <https://scikit-learn.org/>

US_83	Edge Analytics Engine	Data Provider	select a specific set of data to perform analysis	I can get insights regarding data from specific sources, during a specific period.
-------	-----------------------	---------------	---	--

2.6.2.2 Features to be released in the v1.00 version of the platform

All features have been released in the v0.50 version of the platform.

3 CONCLUSIONS AND NEXT STEPS

This document provides a high-level overview of the different components coming out of WP4, described by the DataVaults architecture as components that are working mostly in the data management and analytics tasks of the overall system.

At the time of submitting this document, these components have already progressed with regards to their implementation, following the development plan of the project, as derived from the MVP and the needs of the project. All these requirements should be covered by the Beta release of the project.

As such, the current implemented features are provided to WP5 for the integration tasks, which will also continue towards further developing the features of the different components.

REFERENCES

- [1] DataVaults Consortium, "D4.1 Data Sharing Value Generation and INtelligence Bundles - Version 1," 2021.
- [2] DataVaults Consortium, "D3.2 Security, Privacy and Trust Bundles - Version 2," 2021.