



EU4H-2021-PJ2
EHDS2 Pilot ("HealthData@EU pilot")
101079839

D5.1 **Architecture Definition Document**

21 November 2024



**Co-funded by
the European Union**

Disclaimer: Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HaDEA. Neither the European Union nor the granting authority can be held responsible for them.

0. Document informations

0.1 Authors

Author	Partner
Lionel Grondin	Health Data Hub (France)
Irene Zittlau	Health Data Lab (Germany)

0.2 Reviewers

Reviewer	Partner
Bernd Ahlborn	Health Data Lab (Germany)
Sven Linßen	Health Data Lab (Germany)
Louis Pery	Health Data Hub (France)

0.3 Document history

Date	Version	Editor	Change	Status
20/09/2024	0.1	Lionel Grondin	Initialisation	Draft
26/09/2024	0.2	Bernd Ahlborn Sven Linßen	Comments	Draft
02/10/2024	0.3	Irene Zittlau	Comments	Draft
07/10/2024	0.4	Louis Pery	Comments	Draft
11/10/2024	0.5	Lionel Grondin Irene Zittlau	Changes according to comments	Draft
04/11/2024	0.6	European Commission C1 and R4	Comments	Draft
05/11/2024	0.7	Lionel Grondin Irene Zittlau	Changes according to comments	Draft
21/11/2024	0.8	Lionel Grondin	Minor change	Draft submission for



0.4 Disclaimer

The content of this deliverable represents the views of the author(s) only and is his/her/their sole responsibility; it cannot be considered to reflect the views of the European Commission and/or the Consumers, Health, Agriculture and Food Executive Agency or any other body of the European Union. The European Commission and the Agency do not accept any responsibility for use of its contents.

0.5 Copyright Notice

Copyright © 2022 EHDS2 Consortium Partners. All rights reserved. For more information on the project, please see [the website](https://www.ehds2pilot.eu/)¹.

0.6 Abbreviations

Abbreviation	Definition
API	Application Programming Interface
CEF	Connecting Europe Facility
DCAT-AP	Data Catalog Vocabulary - Application Profile
DG SANTE	Directorate-General SANTE
EHDS	European Health Data Space
HDAB	Health Data Access Body
NCP	National Contact Point
RDBMS	Relational Database Management System
REST-API	Representational State Transfer - Application Programming Interface
TLS	Transport Layer Security
WP5	Work Package 5

¹ <https://www.ehds2pilot.eu/>



0.7 Glossary

Term	Definition
Architecture Definition Document	Final deliverable of Work Package 5
DG SANTE	Department of the European Commission on food safety and health policy topics
European Health Data Space	Health Data Space as part of the Common European Data Spaces
Health DCAT-AP	A health-related extension of the DCAT application profile for sharing information about Catalogues containing Datasets and Data Services descriptions in Europe ²
HealthData@EU	Term for the infrastructure in the EHDS on the secondary use of health data
HealthData@EU Pilot	Project on piloting a first version of the EHDS for secondary use of health data
National Contact Point	
Work Package 5	Technical Work Package within the HealthData@EU Pilot Project focusing on a first IT infrastructure for the EHDS on secondary use of health data

² <https://healthdcat-ap.github.io/>



0.8 Summary table

0. Document informations	2
0.1 Authors	2
0.2 Reviewers	2
0.3 Document history	2
0.4 Disclaimer	3
0.5 Copyright Notice	3
0.6 Abbreviations	3
0.7 Glossary	4
0.8 Summary table	5
1. Introduction	7
1.1 Context	7
1.2 Scope	7
1.3 Objectives of the HealthData@EU infrastructure	7
2. High-Level Design (HLD)	9
2.1 HLD architecture schema	9
2.2 National components	9
2.2.1 Cross Border Gateway	9
2.2.2 National Connector	10
2.2.3 Cross Border Engine	10
2.3 Central platform	10
2.4 Use cases	10
2.4.1 Data discovery : update the EU Dataset Catalogue	10
2.4.2 Data permit : apply for data from the EU Data Access Application form	11
2.4.3 Data permit : update application statuses	12
3. Low-Level Design (LLD) of the National Connector	13
3.1 LLD architecture schema	13
3.2 Description of the services	13
3.2.1 API Gateway	13
3.2.2 AS4 Message Dispatcher	14
3.2.3 Data Discovery	15
3.2.4 Data permit	15
3.2.5 Database	16
3.2.6 Goose	16
3.2.7 Shacl validator	16
3.3 Users	17
3.4 Hosting	17
3.5 APIs	18



3.5.1 Validate datasets	18
3.5.2 Create a dataset	18
3.5.3 Update an existing dataset	18
3.5.4 Delete an existing dataset	19
3.5.5 Restore a catalogue	19
3.5.6 List all applications	20
3.5.7 Update an application	20
3.5.8 Retrieve status of a data discovery message ID	20
3.5.9 Retrieve status of a data permit message ID	21
3.5.10 Create a new API Key on the API Gateway	21
3.5.11 Create a new API Key on the API Gateway	22



1. Introduction

1.1 Context

The European Health Data Space (EHDS) Regulation, adopted in 2024³, aims to establish a secure and structured framework for sharing health data across Europe. Chapter IV of the regulation specifically addresses the reuse or "secondary use" of electronic health data, enabling access for research, innovation, policy-making, and other public health purposes. By fostering interoperability, privacy, and standardisation, EHDS seeks to unlock health data's potential for advancing healthcare and public health while maintaining rigorous data protection and security standards.

The HealthData@EU pilot project is a key component in realising this vision. Its primary objective is to build and test the infrastructure required for seamless, secure data exchange between Member States, facilitating the secondary use of health data in line with EHDS requirements. Through this pilot, the ambition is to set the stage for the HealthData@EU infrastructure that connects National Contact Points (NCPs) to a central platform, enabling harmonised data flows.

1.2 Scope

This *Architecture Definition Document* serve as the final deliverable of Work Package 5 (WP5) on IT Infrastructure within the HealthData@EU pilot project. The document presents a comprehensive vision for the HealthData@EU infrastructure, focusing specifically on the architecture of national components, selected standards, and core technological building blocks.

The infrastructure detailed here supports two primary use cases: **data discovery** and **data permit**. These use cases illustrate key steps in the user journey within the EHDS framework, enabling seamless data sharing and access applications. Additionally, the document details the integration of National Contact Points with the Central Services established by the European Commission, describing the implementation of the National Connector, which acts as the key component for these interactions.


1.3 Objectives of the HealthData@EU infrastructure

The HealthData@EU IT infrastructure aims to facilitate secure, interoperable information exchange between National Contact Points (NCPs) through a unified network, supporting various use cases as outlined by the EHDS Regulation.

The HealthData@EU Pilot experimented, in a Proof Of Concept, an infrastructure that supports the following two use cases :

- Data Discovery, which enables National Contact Points (NCPs) of the network to send dataset descriptions (or metadata) to the central platform which federates these

³ https://ec.europa.eu/commission/presscorner/detail/en/ip_24_2250



received dataset descriptions into a unified European catalogue : the EU dataset catalogue.

- Data Permit, which allows NCPs to receive data access applications for the datasets they are responsible for from data applicants. The applications are filled-in in the European central portal

The high level design of the HealthData@EU infrastructure piloted in this project has been driven by three core principles, namely :

- Scalability: build a distributed network that can easily onboard new contact points or host new use cases
- Open source technologies: leverage transparent and community-driven technologies
- Safety: propose a robust security-oriented solution

The infrastructure utilises the *eDelivery* protocol, an EU CEF building block, to facilitate secure, standardised digital data exchange between nodes, aligning HealthData@EU with EU CEF's broader interoperability and security standards.

As this network is connecting information systems that could be different for each National Contact Point and for each use case, the work package designed a high-level architecture for the National Contact Points made of three core conceptual components (see figure 1) :

- a **Cross Border Gateway** in charge of eDelivery communication with Contact Points and the Central Services,
- a **National Connector** that provides all necessary features to fulfil EHDS regulatory requirements for each step of the user journey,
- and **Cross Border Engines** that allow the National Contact Points to extend the functionalities of their existing information system specifically for the new use cases defined by the regulation.

2. High-Level Design (HLD)

2.1 HLD architecture schema

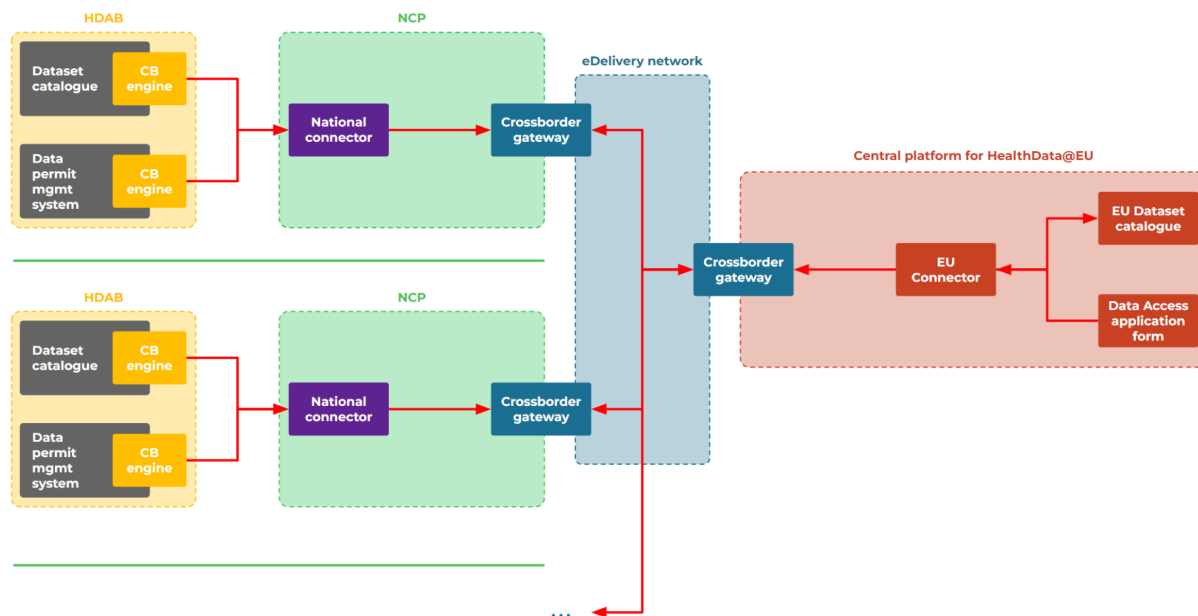


Figure 1 : High-Level Design

2.2 National components

The HealthData@EU infrastructure comprises several key national components, each with a specific role in facilitating secure data exchange between National Contact Points and Central Services.

2.2.1 Cross Border Gateway

A Cross Border Gateway handles the transmission and reception of communications between National Contact Points and the Central Services in a secure and technically standardised manner. It supports the eDelivery protocol. There is one Cross Border Gateway instance for each National Contact Point.

An example of a use case for the Cross Border Gateway is to send an acceptance message of a 'data permit request' from the National Contact Points' Cross Border Gateway to the Central Services' Cross Border Gateway through eDelivery.

The HealthData@EU pilot project uses '[Domibus](https://ec.europa.eu/digital-building-blocks/sites/display/DIGITAL/Domibus)⁴' for the Cross Border Gateway. Domibus is an

⁴ <https://ec.europa.eu/digital-building-blocks/sites/display/DIGITAL/Domibus>

open source reference implementation of the eDelivery access point, maintained by the European Commission.

2.2.2 National Connector

The National Connector is a reference implementation of a software component exposing capabilities that are common to all National Contact Points to fulfil the user journey steps. Each National Contact Point deploys and runs one National Connector. The main task of the National Connector is to add an abstraction layer before the Cross Border Gateway and therefore to expose specific features needed to fulfil the infrastructure use cases.

An example of a use case for the National Connector is to ensure that an acceptance message of a 'data permit request' by the National Contact Point is correctly received and understood by the Central Services.

2.2.3 Cross Border Engine

Cross Border Engines are extensions of the Health Data Access Bodies' (HDABs) information systems (e.g. dataset catalogues or data permit management systems) that add new functionalities to the systems in order to enable cross border use cases. There can be one Cross Border Engine instance for each application of an HDAB, e.g. one Cross Border Engine instance for the National Dataset Catalogue.

An example of a use case for the Cross Border Engine is to automatically identify when there is a change on the national dataset catalogue and call the APIs of the National Connector to push the changes on the EU Dataset Catalogue.

2.3 Central platform

The central platform of HealthData@EU is developed and maintained by DG SANTE. For now it comprises an EU Dataset Catalogue and an EU Data Access application form. The detailed description of the functionalities and the respective architecture are out of scope of this document.

2.4 Use cases

2.4.1 Data discovery : update the EU Dataset Catalogue

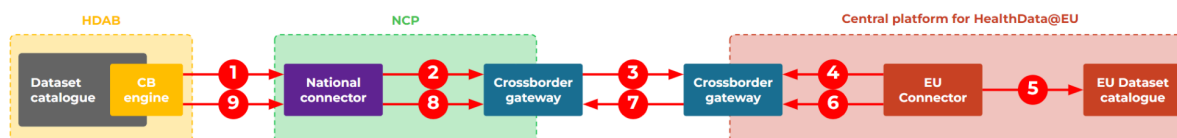


Figure 2 : Workflow to update the EU Dataset Catalogue

- (1) The Cross Border Engine is in charge of identifying when there is a change on the national dataset catalogue. It then calls the APIs of the National Connector with the description of the dataset in DCAT-AP.
- (2) For data discovery, the National Connector exposes four REST-APIs (create, update, delete, restore). When one of those APIs is called, the National Connector prepares the eDelivery message (encoding to base64, authentication to eDelivery, ...) and sends it to the National Cross Border Gateway.
- (3) The eDelivery message is sent over the network to the central platform Cross Border Gateway, leveraging all security features of eDelivery (encryption, authentication, ...).
- (4) The EU connector reads all eDelivery messages.
- (5) Once a new data discovery message is received, it performs the associated action on the EU Dataset catalogue (create, update, delete, restore).
- (6) Depending on the success of the operation, the EU connector prepares the eDelivery message (encoding to base64, authentication to eDelivery, ...) containing the status of the operation and sends it to the central platform Cross Border Gateway.
- (7) The eDelivery message is sent over the network to the national Cross Border Gateway, leveraging all security features of eDelivery (encryption, authentication, ...).
- (8) That status of the operation is stored in the database of the National Connector.
- (9) The Crossborder Engine can check the status by calling an API endpoint of the National Connector. If it fails it can raise an alert to the end user managing the National Dataset Catalogue.

2.4.2 Data permit : apply for data from the EU Data Access Application form



Figure 3 : Workflow to apply for data from the EU Data Access Application form

- (1) When a new user fills in a data access application form and clicks on submit, the values are sent to the EU connector through a REST-API call. The EU Connector then prepares the eDelivery message (encoding to base64, authentication to eDelivery, ...) and sends it to the Central Platform Cross Border Gateway.
- (2) The eDelivery message is sent over the network to the national Cross Border Gateway, leveraging all security features of eDelivery (encryption, authentication, ...).
- (3) For the data permit, the National connector stores each Data Access Application in a database. It also exposes a REST-API (GET) to read all applications.
- (4) The Cross Border Engine is in charge of regularly calling the GET-API of the National Connector to import new Data Access Applications into the National Data Permit Management System.

The current implementation for this use case does not yet support a confirmation workflow after the application is received on the national side. This is foreseen as a feature to be added in future releases.

2.4.3 Data permit : update application statuses

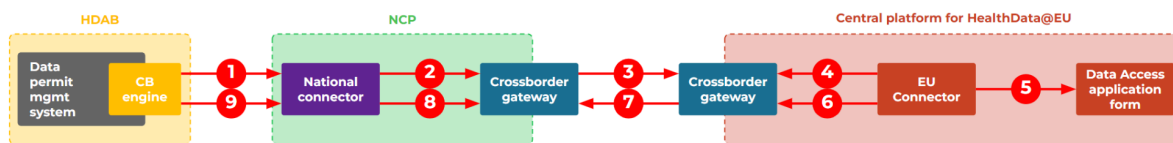


Figure 4 : Workflow to update application statuses

- (1) The Cross Border Engine is in charge of identifying when there is a change in the status of the National Data Permit Management System. It then calls the APIs of the National Connector with the updated status.
- (2) For the data permit, the National Connector exposes a REST-API to update the status. When the API is called, the National Connector stores the update in the local database and prepares the eDelivery message (encoding to base64, authentication to eDelivery, ...) and sends it to the national Cross Border Gateway.
- (3) The eDelivery message is sent over the network to the Central Platform Cross Border Gateway, leveraging all security features of eDelivery (encryption, authentication, ...).
- (4) The EU Connector reads all eDelivery messages.
- (5) Once a new data permit message is received, it performs the associated action in the EU Portal.
- (6) Depending on the success of the operation, the EU connector prepares the eDelivery message (encoding to base64, authentication to eDelivery, ...) containing the status of the operation and sends it to the central platform Cross Border Gateway.
- (7) The eDelivery message is sent over the network to the national Cross Border Gateway, leveraging all security features of eDelivery (encryption, authentication, ...).
- (8) That status of the operation is stored in the database of the National Connector.
- (9) The Crossborder Engine can check the status by calling an API endpoint of the National Connector. If it fails it can raise an alert to the end user managing the national Data Permit Management System.

3. Low-Level Design (LLD) of the National Connector

3.1 LLD architecture schema

The decision to design the application following a microservices architecture was driven by several key advantages: Microservices offer a modular approach, allowing for the independent development, deployment, and scaling of individual services, which significantly enhances flexibility and maintainability. Moreover, the architecture promotes a clear separation of concerns, enabling teams to work on different components in parallel, reducing development time and minimising bottlenecks. Therefore, this architecture results in greater resilience as the failure of one service does not impact the entire system, improving overall fault tolerance.

Each microservice consists of a docker container. The orchestration of all containers can either be implemented with 'Docker Compose' or 'Kubernetes' (cf section 3.4)

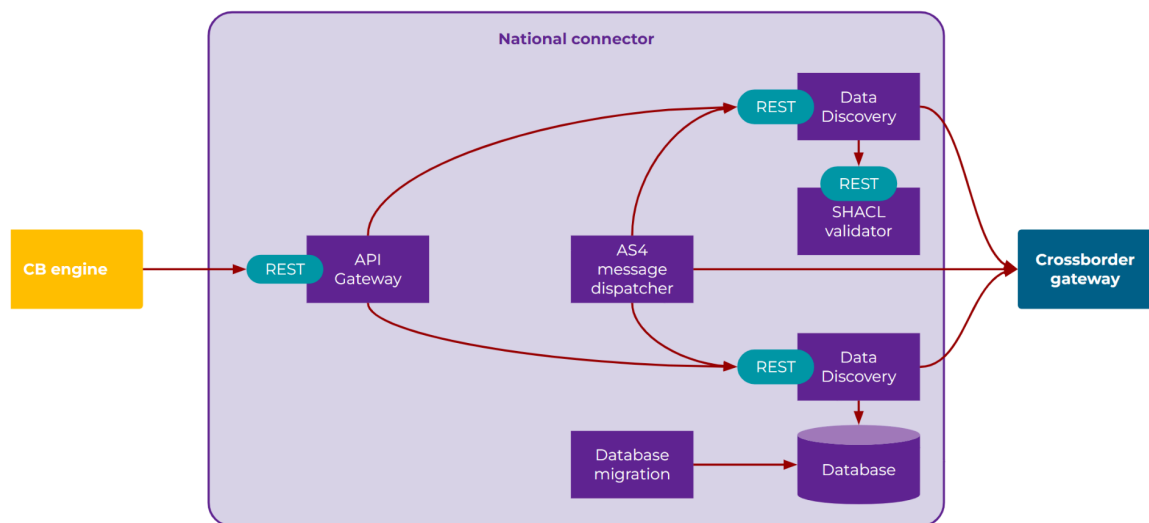



Figure 4 : Low-Level Design of the National Connector

3.2 Description of the services

3.2.1 API Gateway

The API Gateway is a crucial component in this microservices architecture that acts as a single entry point for client requests to all other backend services. Its main role is to route the requests to the correct backend service, ensuring that clients do not need to interact directly with each individual service. By consolidating these interactions, the API Gateway simplifies client communication and enhances security by providing centralised authentication. Additionally, in order to address further needs, it could also manage load balancing, caching,



and monitoring; improving overall system performance.

Its role is the generic routing of requests while it can easily be replaced by any API Gateway used by a National Connector. This would allow a better integration with the rest of the information system of the National Contact Point.

The API Gateway has been developed in 'Go' and the code is [open source](#)⁵.

The functionalities provided by this service are :

Functionality ID	Functionality
FT01	Provide a single entry point for client requests
FT02	Route the requests to the correct microservices
FT03	Provide TLS encryption endpoint for incoming requests
FT04	Verify the authentication of incoming requests

3.2.2 AS4 Message Dispatcher

The AS4 message dispatcher is in charge of reading incoming messages from the Cross Border Gateway of the National Contact Point and to dispatch it to the other microservices of the National Connector (data permit or data discovery) for the further treatment of that message.

The dispatcher requires a Domibus-compatible Cross Border Gateway. If a National Contact Point does not use Domibus as their Cross Border Gateway, this component needs to be replaced by another custom implementation that is compatible with the Cross Border Gateway implementation.

The AS4 message dispatcher has been developed in 'Go' and the code is [open source](#)⁶.

The functionalities provided by this service are the following :

Functionality ID	Functionality
FT04	Pull incoming eDelivery messages received on the Cross Border Gateway (Domibus) and dispatch the message to other services of the National Connector

⁵ <https://code.europa.eu/healthdataeu-nodes/hdeupoc/-/tree/main>

⁶ <https://code.europa.eu/healthdataeu-nodes/hdeupoc/-/tree/main>

3.2.3 Data Discovery

The data discovery service is in charge of handling all the business logic related to data discovery use cases. It prepares and builds the eDelivery message for each operation implemented : creation, update and deletion of datasets in the EU Dataset Catalogue and restoration of a catalogue.

The Data Discovery Service has been developed in 'Go' and the code is [open source](#)⁷.

The functionalities provided by this service are the following :

Functionality ID	Functionality
FT05	Build eDelivery messages for creation of metadata
FT06	Build eDelivery messages for update of metadata
FT07	Build eDelivery messages for deletion of metadata
FT08	Build eDelivery messages for restoration of catalogues
FT09	Store and provide status of data discovery operations sent over eDelivery

3.2.4 Data permit

The data permit service is in charge of handling all the business logic related to data permit use cases. It prepares and builds the eDelivery message for each operation implemented : Retrieve applications received by eDelivery and store them in the database, communicate information about a single application and prepare and build the eDelivery message for a status update of an application.

The Data Permit service has been developed in 'Go' and the code is [open source](#)⁸.

The functionalities provided by this service are the following :

Functionality ID	Functionality
FT10	Retrieve applications received by eDelivery and store them in the database
FT11	Communicate information about a single application
FT12	Build eDelivery messages for update of a status of an application

⁷ <https://code.europa.eu/healthdataeu-nodes/hdeupoc/-/tree/main>

⁸ <https://code.europa.eu/healthdataeu-nodes/hdeupoc/-/tree/main>

FT13	Store and provide status of data permit operations sent over eDelivery
------	------------------------------------------------------------------------

3.2.5 Database

In the National Connector, the database is used to store information such as incoming data permit applications. 'PostgreSQL', commonly referred to as Postgres, has been used during the pilot. Postgres is a relational database management system (RDBMS) known for its robustness, flexibility and standards compliance.

PostgreSQL is open source.

The functionalities provided by this service are the following :

Functionality ID	Functionality
FT14	Store data permit applications

3.2.6 Goose

'Goose' is a lightweight and flexible database migration tool designed for managing schema changes in PostgreSQL.

In the National Connector, Goose is used to set up the database when the application is started for the first time and allow migration to new versions of the application.

Goose is open source.

The functionalities provided by this service are the following :


Functionality ID	Functionality
FT15	Initialise the database at initial setup
FT16	Migrate the database schema in case of new version

3.2.7 Shacl validator

The 'shacl validator service' is in charge of validating the compliance of a dataset of metadata in regard to the DCAT-AP specification. This operation is optional in the workflow. However, at the target, the DCAT-AP specifications should be replaced by Health DCAT-AP specifications.

Shacl validator is open source.

The functionalities provided by this service are the following :



Functionality ID	Functionality
FT17	Provide a compliance status of a dataset metadata against dcat-ap specifications

3.3 Users

The National Connector is not meant to be used by end users. The clients that will consume the APIs can authenticate themselves by using an API key of the API Gateway.

A first version of API keys management has been implemented during the pilot and would require further development before production use in order to meet state of the art security requirements.

Currently, at initial setup, all API endpoints can be consumed without any authentication mechanism. API keys can be created using a dedicated API endpoint. When a first API key is created, authentication is then enforced when consuming any API endpoint, including creation of other API keys. It is strongly recommended to create an API key immediately when deploying the application.

The deletion of all API keys can be done by calling a dedicated API endpoint. This operation requires to be authenticated. All API keys are removed and, therefore, all API endpoints can again be consumed without authentication.

3.4 Hosting

As the National Connector will be deployed and maintained by different IT teams with different skills, the same microservices design can be deployed either with 'Docker Compose' or 'Kubernetes'.

When deploying a microservices application, Docker Compose provides a simple, straightforward way to manage and run containers on a single host, making it ideal for development environments or small-scale deployments. It is easy to configure, with a focus on simplicity and quick setup, but it lacks advanced orchestration features.

The other option, Kubernetes, is designed for large, distributed systems, offering robust features like auto-scaling, self-healing, service discovery, and seamless rolling updates. Kubernetes excels in managing complex microservices architectures across multiple nodes, making it the better choice for production environments where scalability, reliability, and orchestration are critical. However, it comes with a steeper learning curve and more operational overhead compared to Docker Compose.

3.5 APIs

All APIs described in this section are also documented with Open API syntax. [This documentation](#)⁹ is available on the repository of the HealthData@EU Pilot.

3.5.1 Validate datasets

Path	/data-discovery/datasets/validate	
Method	POST	
Description	Validate a dataset with SHACL API	
Parameters	None	
Request body	application/rdf	
Responses	200	Validation success
	400	Invalid input
	500	Internal error

3.5.2 Create a dataset

Path	/data-discovery/datasets	
Method	POST	
Description	Create a dataset to the Central service through Domibus	
Parameters	None	
Request body	application/rdf	
Responses	202	Creation accepted
	400	Invalid input
	500	Internal error

3.5.3 Update an existing dataset

Path	/data-discovery/datasets
------	--------------------------

⁹ <https://code.europa.eu/healthdataeu-nodes/hdeupoc/-/wikis/National-Connector/WebAPI/OpenAPI>

Method		Put
Description		Update an existing dataset on the Central service through Domibus
Parameters		None
Request body		application/rdf
Responses	202	Update accepted
	400	Invalid input
	500	Internal error

3.5.4 Delete an existing dataset

Path		/data-discovery/datasets
Method		DELETE
Description		Delete an existing dataset by dct identifier
Parameters	dct-id	Unique identifier of a dataset
Request body		None
Responses	202	Delete accepted
	400	Invalid dct identifier
	500	Internal error

3.5.5 Restore a catalogue

Path		/data-discovery/catalogs/restore
Method		POST
Description		Restore a full catalogue on the EU catalogue, by deleting all previous dataset and creating all new datasets. This foreseen feature has not yet been implemented in the EU catalogue.
Parameters		None
Request body		application/json

Responses	202	Restore accepted
	400	Invalid input
	500	Internal error

3.5.6 List all applications

Path	/data-permit/applications	
Method	GET	
Description	Fetch all applications	
Parameters	None	
Request body	None	
Responses	200	Successful returning all applications
	500	Internal error

3.5.7 Update an application

Path	/data-permit/applications/{application-id}	
Method	PUT	
Description	Update the status of a specific application	
Parameters	application-id	application ID to update
Request body	application/json	
Responses	202	Update accepted
	400	Invalid input
	500	Internal error

3.5.8 Retrieve status of a data discovery message ID

Path	/data-discovery/track-messages/{message-id}	
Method	GET	

Description		Request returns information about the status of a specific request
Parameters	message-id	ID of the message
Request body		None
Responses	200	Return the status of the message
	400	Invalid message ID
	500	Internal error

3.5.9 Retrieve status of a data permit message ID

Path		/data-permit/track-messages/{message-id}
Method		GET
Description		Request returns information about the status of a specific request
Parameters	message-id	ID of the message
Request body		None
Responses	200	Return the status of the message
	400	Invalid message ID
	500	Internal error

3.5.10 Create a new API Key on the API Gateway

Path		/users
Method		POST
Description		Create a user and generating an API Key
Parameters		None
Request body		application/json
Responses	201	User created successfully
	400	Invalid input



	500	Internal error
--	-----	----------------

3.5.11 Create a new API Key on the API Gateway

Path	/users	
Method	DELETE	
Description	Delete all users	
Parameters	None	
Request body	None	
Responses	204	All users has been deleted
	500	Internal error