COMFORTage

Prediction, monitoring and personalized recommendations for prevention and relief of dementia and frailty

Working Paper

COMFORTage contribution to Ageing-EHDS Data Space

Coordinated by Ubitech and Leanxcale

This document is part of a project that has received funding from the European Union's Horizon Europe programme under agreement 101137301 — COMFORTAGE HORIZON-HLTH-2023-STAYHLTH-01.
The content of this document reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

Funded by the European Union

Grant Agreement number: 101137301 — COMFORTAGE HORIZON-HLTH -2023-STAYHLTH-01

Definitions, Acronyms and Abbreviations

Abbreviation	Description
2FA	Two-Factor Authentication
AI	Artificial Intelligence
API	Application Programming Interface
CA	(COMFORTage) Consortium Agreement
DIH	Digital Innovation Hub
EHDS	European Health Data Space
GA	Grant Agreement
HDAB	Health Data Access Bodies
IAM	Identity and Access Management
ICML	Integrated Care Model Library
IDSA	International Data Spaces Association
LL	Living Lab
OTP	One-time Based Password
PM	Policy Manager
RWD	Real World Data
SPEs	Secure Processing Environments
SSO	Single Sign-On
TOTP	Time-Based One-Time Passwords
VHP	Virtual Healthcare Platform

Funded by
the European Union

Driving Innovation in Secure and Interoperable Health Data Spaces: The COMFORTage Ageing-EHDS Approach

Contents

1 Executive Summary

Europe is experiencing an unprecedented demographic transformation, with populations ageing at a pace unmatched in history. As longevity increases, so too does the incidence of chronic diseases, multimorbidity, dementia, and frailty, leading to increased pressure on healthcare systems ill-equipped to handle these multi-factorial challenges. The traditional healthcare paradigm — episodic, siloed, and often reactive — fails to provide the holistic, continuous, and preventative approaches necessary to ensure healthy ageing.

Against this backdrop, the COMFORTage project emerges as a groundbreaking initiative aimed at transforming how Europe approaches ageing-related health challenges. It leverages cutting-edge technological innovations, advanced data governance models, and human-centric design principles to build an integrated digital ecosystem capable of supporting early detection, personalised interventions, and secure, cross-border data sharing.

This working paper captures the key architectural, regulatory, and organisational innovations underpinning the COMFORTage Ageing-EHDS platform. By aligning directly with the European Health Data Space (EHDS) framework and embedding the International Data Spaces Association (IDSA) Reference Architecture Model (RAM), COMFORTage lays the groundwork for scalable, interoperable, privacy-preserving data infrastructures. Its core components — from multi-realm Identity and Access Management (IAM) and advanced policy-based access controls to IDSA-compliant data connectors and Secure Processing Environments (SPEs) — together form a blueprint for future European health data spaces.

Through detailed exploration of COMFORTage's architecture, pilot implementations, lessons learned, and strategic outlook, this paper provides critical insights into how innovation and compliance can coalesce to deliver tangible benefits for older adults, caregivers, clinicians, researchers, and policymakers. It serves as a reference for similar initiatives aiming to align health data infrastructures with the ambitious goals of the EHDS while maintaining the highest standards of data sovereignty, security, and patient trust.

2 Introduction: The Challenge and the Vision

2.1 The demographic imperative

Europe's demographic landscape is shifting rapidly. Projections indicate that by 2050, nearly one in three Europeans will be over the age of 65, with the share of people over 80 doubling. These trends come with profound healthcare implications, particularly concerning the surge in age-related conditions such as dementia, frailty, osteoporosis, and cardiovascular diseases.

Dementia alone is estimated to affect over 14 million Europeans by 2050, imposing immense personal, social, and economic costs. Traditional healthcare models are fundamentally ill-suited to address these complex, multi-dimensional challenges. They often lack the ability to integrate disparate data sources, predict risks before clinical manifestations, and provide continuous, personalised support adapted to evolving patient needs.

2.2 The technological opportunity

Simultaneously, the digital revolution offers unprecedented opportunities to rethink ageing care. We refer in particular to following technologies:

- **Wearable sensors and smart home technologies** which can capture real-time data on mobility, cognition, nutrition, and social engagement.
- **Genomic profiling and molecular diagnostics** which enable deeper insights into individual disease risks.
- **AI-driven models** which can synthesise these multi-modal data streams to generate personalised risk scores, predict disease trajectories, and recommend tailored interventions.

However, realising these opportunities requires robust data infrastructures capable of securely integrating, harmonising, and analysing sensitive personal health data — all while safeguarding individual privacy, maintaining data sovereignty, and building public trust.

2.3 The EHDS as a regulatory catalyst

Recognising both the promise and the risks, the European Commission has launched the European Health Data Space (EHDS) initiative. The EHDS seeks to create an EU-wide framework enabling:

- **Primary use:** seamless, secure sharing of health data across borders for treatment continuity.
- **Secondary use:** ethically governed access to aggregated data for research, innovation, public health, and policy formulation.

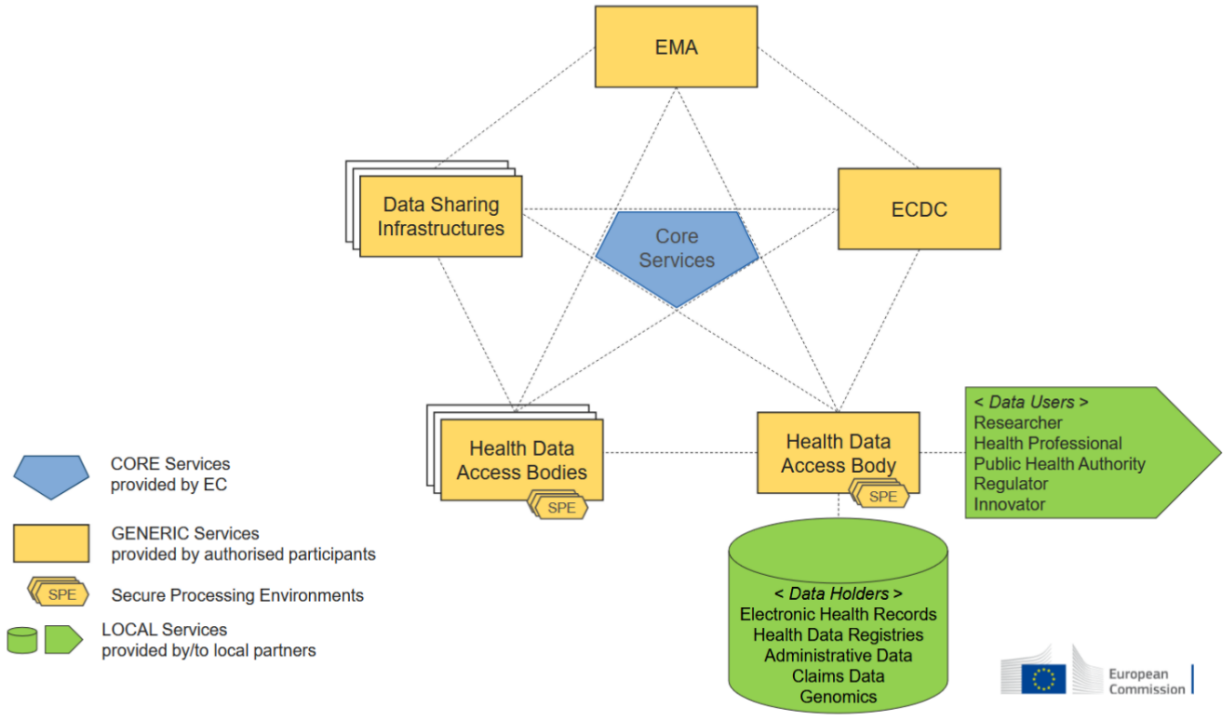


Figure 1: High-Level Architecture in the proposed EHDS regulation

The EHDS mandates strong safeguards, including explicit consent mechanisms, Secure Processing Environments (SPEs), and Health Data Access Bodies (HDABs) to oversee secondary use requests. For projects like COMFORTage, aligning early with these specifications offers a strategic advantage, ensuring both future compliance and readiness for integration into broader European data ecosystems. COMFORTage does not explicitly reference SPEs. However, its design appears compatible with SPE principles, such as local data control, algorithmic processing under governance constraints, and minimal external exposure of personal data.

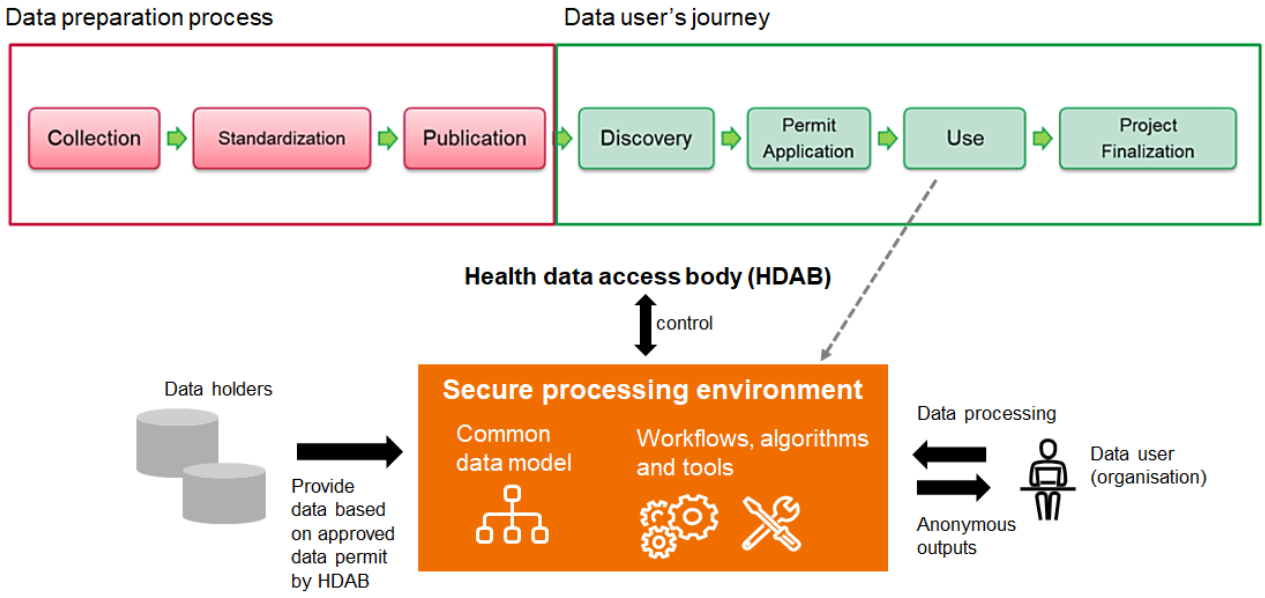


Figure 2: Secure Processing Environment in EHDS

2.4 The COMFORTage vision

COMFORTage stands at this critical intersection of demographic necessity, technological possibility, and regulatory transformation. Its vision is to create a comprehensive platform that:

- Empowers older adults to actively manage their health through personalised digital tools.
- Enables clinicians and clinical researchers to make data-driven, predictive decisions based on holistic patient profiles.
- Provides researchers with ethically accessible, harmonised datasets for advancing dementia and frailty science.
- Aligns meticulously with GDPR, upcoming AI regulations, and the EHDS to ensure security, privacy, and societal acceptance.

While the official specifications of the EHDS are currently in the making, by embedding IDSA principles of data sovereignty and federated governance, COMFORTage not only builds a platform for today's needs but also future-proofs its architecture for the evolving European data landscape.

3 The COMFORTage Innovation Framework

3.1 Four pillars of integrated innovation

At its core, COMFORTage employs a holistic innovation model spanning **four complementary domains**, each critical to addressing the multifactorial nature of ageing-related health risks.

Medical innovations

COMFORTage advances precision prevention for ageing-related conditions through:

- **Longitudinal cohorts and digital biomarkers:** Using smart wearables and home-based sensors to capture subtle changes in gait, speech, and cognition that precede clinical diagnoses.
- **Risk stratification models:** Integrating clinical data (e.g. blood pressure, cholesterol), behavioural data (e.g. daily activity levels), and genetic profiles to build predictive algorithms for dementia and frailty progression.

AI and data science innovations

The project develops state-of-the-art AI capabilities including:

- **Explainable machine learning models:** To ensure clinicians can understand, interrogate, and trust risk scores or intervention recommendations.
- **Digital twins of ageing trajectories:** Virtual representations of individual patients that simulate the impact of interventions, enabling personalised decision-making.

Social and behavioural innovations

Recognising that technology must serve people, COMFORTage embeds:

- **Living Labs:** Real-world environments where older adults, caregivers, clinicians, and developers collaboratively design and iteratively test digital tools.
- **Digital Innovation Hubs (DIHs):** Networks that support scaling of validated solutions, training local actors, and fostering regional ecosystems.
- **Behavioural engagement strategies:** Tailored coaching interfaces and educational modules to promote digital literacy and sustained participation.

Data governance and privacy innovations

COMFORTage goes beyond compliance by:

- Designing privacy as an architectural feature, not an afterthought.
- Implementing multi-realm IAM with embedded consent tokens.
- Aligning with IDSA RAM to enforce data sovereignty through decentralised, contract-governed exchanges.

This integrative approach ensures that COMFORTage is not simply building a platform, but a socio-technical ecosystem responsive to the complexities of ageing care.

4 Strategic Alignment with EHDS and IDSA

4.1 The European Health Data Space: a new regulatory paradigm

The EHDS represents a landmark policy shift, seeking to build:

- **A common EU framework for primary use**, where citizens can seamlessly access and share their health data across borders for clinical care.
- **A structured ecosystem for secondary use**, enabling research, innovation, public health monitoring, and policy development.

Under the proposed EHDS regulation:

- **Health Data Access Bodies (HDABs)** will vet and authorise secondary use requests.
- **Secure Processing Environments (SPEs)** will become mandatory for processing sensitive health data for research or policy analytics.
- Citizens will benefit from **digital control dashboards**, enhancing transparency and empowering dynamic consent choices.

COMFORTage's architecture explicitly anticipates these requirements by:

- Implementing transparent consent dashboards within its VHP interfaces.
- Building contractual data exchange mechanisms that align with upcoming HDAB oversight processes.

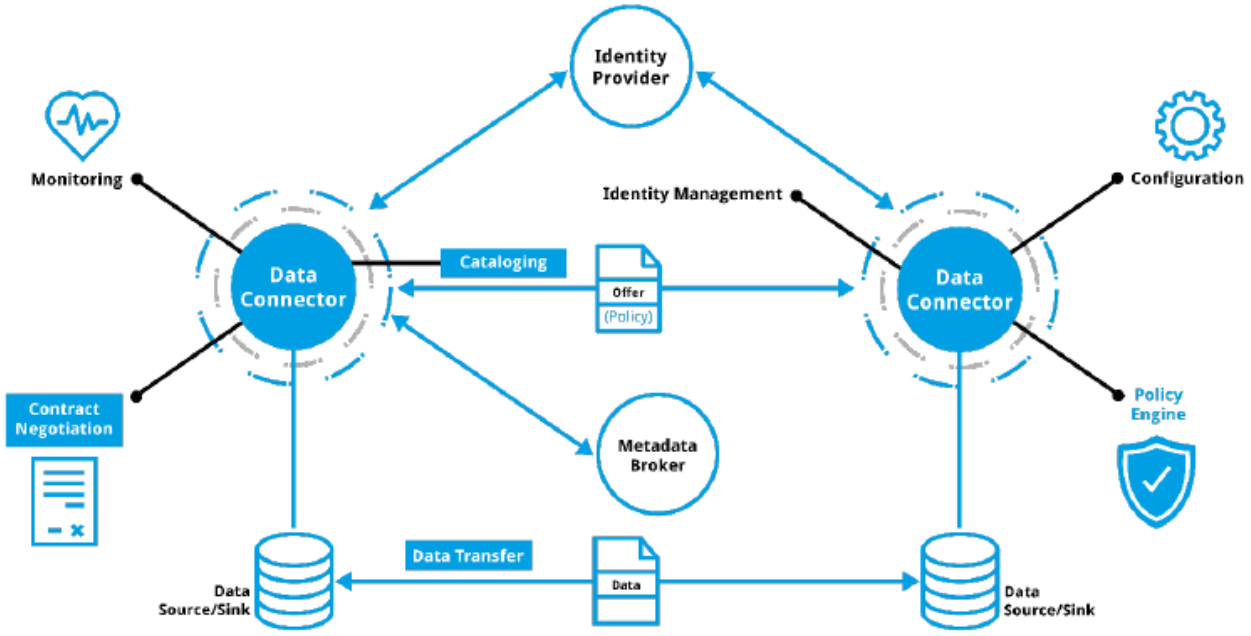
4.2 IDSA: embedding data sovereignty and federated trust

The International Data Spaces Association (IDSA) Reference Architecture Model (RAM) provides the federated governance layer that complements EHDS objectives. While the EHDS focuses on rights and regulatory oversight, IDSA operationalises these through:

- **Data sovereignty guarantees:** Ensuring that data owners control how, when, and under what conditions their data is used, even post-transfer.
- **Decentralised trust mechanisms:** Using certification, policy enforcement points, and clearing houses to build trust without central monopolies.
- **Usage contracts embedded into transactions:** Each data exchange is governed by digitally signed contracts specifying permissible processing operations.

COMFORTage's infrastructure integrates IDSA connectors to:

- Enforce policies directly at the data exchange layer.
- Maintain immutable logs of every transaction via clearing house services.
- Facilitate cross-organisational data flows under explicit contractual terms, paving the way for future pan-European health data collaborations.



- Figure 3: Data Exchange Services realized by an IDSA compliant data connector

4.3 Use cases of EHDS

A concrete use case can be created to demonstrate how a data researcher can benefit from the EHDS, and thus showcasing the advancements that are foreseen to be provided by the COMFORTage project. Initially, we consider a group of data scientists and researchers that want to collaborate for the development of novel innovative health data applications, using the available data harmonised in a common data model, and probably some of the key **Integrated Care Model Library** (ICML) functionalities of the COMFORTage project. They would need to use EHDS data resources and SPEs to develop their envisioned solution. EHDS can be exploited to provide the default mechanisms for accessing health data for scientific research and development purposes, being able to access large data resources spanning several countries.

In a traditional manner, the group of data scientists and researchers should have been able to conduct a **preliminary research** in order to identify and discover available datasets of the respective data owners. Once the agreements are being signed related with the data usage, transfer and liability, the data collected from the data owner systems will need to be further transferred to local machines on the researchers' computers. There is where data will need to be analysed with the available resources provided by the research team or the individual researchers. Once the processing is finished after the end of the project, the data needs to be deleted according to the agreements signed by the data owner and the data processors. This means that the researchers are responsible for the deletion of the data.

Using the EHDS principles, now the data scientists and researchers can search and discover potential datasets of interest through the EHDS data catalogue. Once an application is filled to the EHDS system about the selected data of interest, the HDAB processes the agreement and eventually requests and combines the data from the related data owners. The HDAB is now responsible to transfer the pseudo-anonymized datasets to the selected SPE that grants access to the named researchers. After this procedure, data is processed on the authorised SPE, while potential results and software artefacts or AI models can be further exported through the HDAB. The latter is also responsible to delete the corresponding data and SPE once the project ends, facilitating data sovereignty.

5 Overall architecture of the COMFORTage platform

Figure 4 represents the overall architecture of the COMFORTage platform.

The green boxes highlight the main components and their position in the overall architecture. The components under the Authentication and Authorisation block play a critical role in managing user identities, enforcing access policies, and safeguarding sensitive health data. Meanwhile, the Dataspace Connectors enable seamless data integration from various sources, including internal systems, emerging data sources, and user-controlled Opt-in Tools.

The captured data will then be harmonised into a format known as Holistic Healthcare Records (HHRs) through the HHR Harmonisation component and eventually stored in the IKB. These components are integral to the COMFORTage architecture as they will operate as the starting point for providing comprehensive and reliable data for analysis and decision-making.

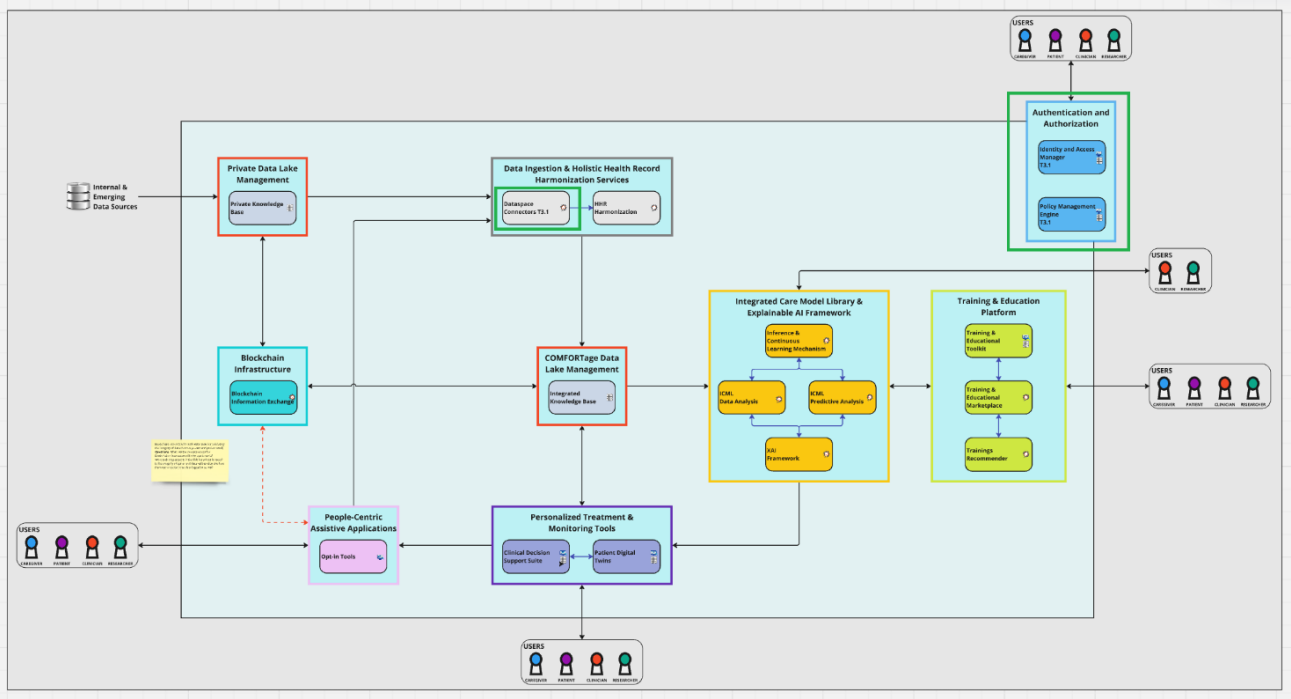


Figure 4: COMFORTage Architecture

6 Architectural Innovation: IAM, Consent, and Privacy Engineering

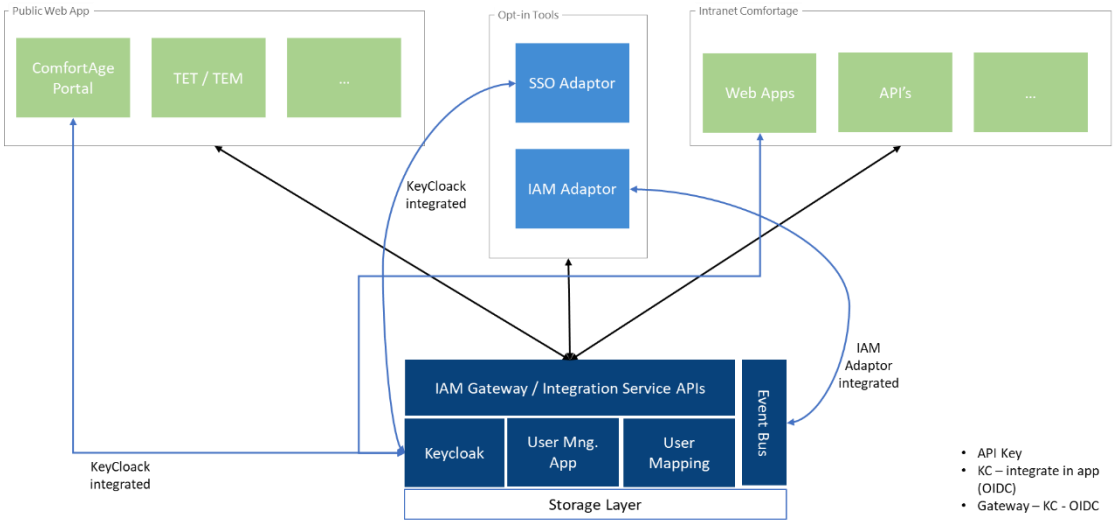
6.1 Multi-realm IAM for secure and scalable studies

A standout architectural feature of COMFORTage is its **multi-realm Identity and Access Management (IAM)** built on Keycloak. This setup enables:

- **Strict logical separation:** Each pilot study, clinical site, or Living Lab operates within its own IAM realm, complete with isolated user directories, role hierarchies, and policy sets.
- **Scalability:** Thousands of users — patients, caregivers, clinicians, researchers — can be onboarded without cross-contamination of permissions or data access.
- **Tailored governance:** Different pilots can enforce specific national or institutional privacy requirements without constraining other realms.

Administrators benefit from intuitive dashboards for managing:

- User onboarding and multi-factor authentication setup.
- Role assignments (patient, clinician, caregiver, researcher, admin).
- Fine-grained policy definitions that adjust based on evolving consent or regulatory needs.



- Figure 5: IAM high level architecture

6.2 From RBAC to PBAC: embedding dynamic, context-aware controls

While most healthcare IAM systems rely solely on Role-Based Access Control (RBAC), COMFORTage advances this by integrating **Policy-Based Access Control (PBAC)**, allowing:

- Contextual checks such as time-bound access (e.g. post-discharge follow-ups), organisational membership, or purpose-of-use tags.
- Real-time evaluation of whether consent attributes within JWTs permit specific data disclosures.

For example, a researcher querying anonymised datasets through an IDSA connector might trigger a PBAC policy that checks:

- Whether the individual consent token explicitly allows research use.
- Whether the dataset falls under the contractual scope approved by the HDAB.
- Whether time constraints (e.g. consent valid only until 2025) are respected.

6.3 Consent-infused JWTs for real-time compliance

COMFORTage ensures that consent is not a static checkbox recorded at onboarding, but a dynamic, enforceable feature embedded into the authentication fabric. This is achieved by:

- Generating JWTs (JSON Web Tokens) at login or session refresh that encode consent parameters directly in the payload.
- Having each microservice and IDSA connector validate these tokens before serving requests.

Example simplified JWT payload:

json
CopierModifier

{
"sub": "u1_pilot3",
"roles": ["patient"],
"consent": {
"data_sharing": true,
"ai_recommendations": false,
"research_use": true
},
"iat": 1710301800,
"exp": 1710388200
}

Thus, if a patient withdraws consent for AI coaching modules, their subsequent JWTs will automatically reflect "ai_recommendations": false, blocking such services without additional manual intervention.

This architecture transforms compliance from a bureaucratic overlay into a **programmatically enforced reality**, significantly reducing legal risks and fostering trust.

7 Data Source Connectors (Connectors for Contributing Data)

This section is dedicated to the **data source connectors** of the integrated COMFORTage platform. The scope of these connectors is to acquire data coming from a variety of heterogeneous external sources, in order to feed the relevant **data ingestion pipeline**. The objective of the latter is to capture, clean, harmonise and finally provide the input data to the **IKB** of the platform, under the standardised HHR model. The purpose of the *data source connectors* is to provide the means to the *data ingestion pipelines* to acquire the raw data, in a unified manner by encapsulating the deep details of the specific external data source provider.

This section will firstly provide the overview of the **data source connectors**, followed by a deeper technical description of its design decisions and implementation. Then, the documentation of this component is provided, describing the relevant APIs, prerequisites and installation guides, user guides and next steps for future development.

7.1 Overview

The **data source connectors** are an integral part of the overall **data ingestion pipelines** of the integrated COMFORTage solution. The **data ingestion pipelines** themselves can be depicted in the COMFORTage reference architecture, under the **Data Ingestion** *and* **Holistic Health Records Harmonisation** building box, as shown in figure 4.

The data ingestion pipelines are meant to retrieve input data from a variety of heterogeneous sources and provide output data to the **COMFORTage Data Lake Management**, and more precisely, to the **Integrated Knowledge Base** that lies within. From the reference architecture, we can distinguish two conceptual types of input resources: raw data, coming from a conceptual private data lake, and data coming from the **People-Centric Assistive Applications** and the corresponding opt-in tools. The first category concerns data that are being provided by external data providers (i.e.

hospitals, medical centres, etc.), while the second category concerns data that are being generated from the user interaction with the opt-in tools. As a fact, the purpose of the **data source connectors** is to provide common way to retrieve data from these heterogeneous sources.

Going a step deeper into the **Data Ingestion and Holistic Health Records Harmonisation** building box, the following figure depicts where the **data source connectors** lie into the overall data ingestion pipelines.

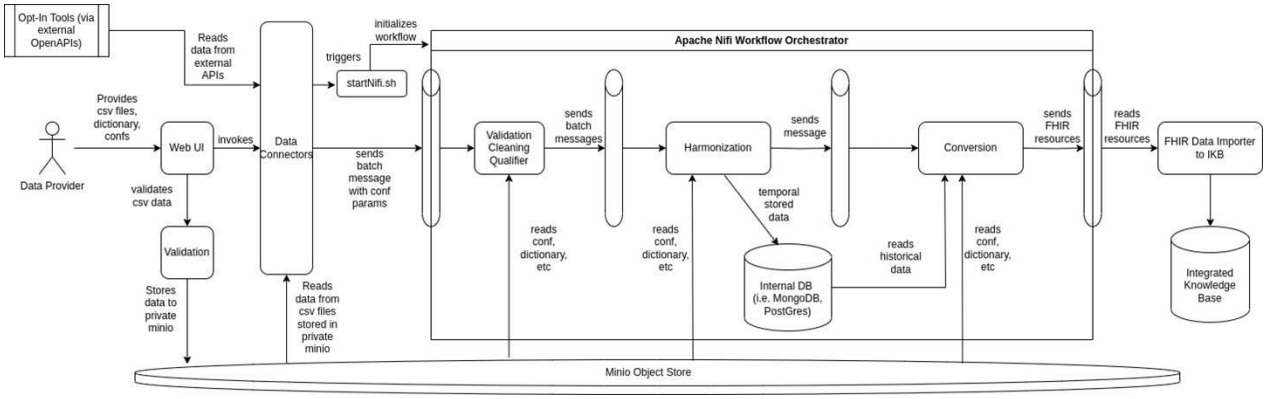


Figure 6: Data Ingestion Pipelines Diagram

As depicted in this diagram, the data providers communicate via a web interface in order to setup the corresponding data ingestion pipeline. There, they need to provide specific information like data specifications (i.e. the dictionary of the data themselves), where the data can be actually retrieved (i.e. from a local file or from an external OpenAPI in the case of opt-in tools, etc.). If the data are meant to be provided as separate files, the data providers also need to upload the corresponding files, that after being validated against the dictionary, they are stored into a distributed file system, which plays the role of the private data lake of the COMFORTage reference architecture. In our implementation, we decided to make use of the MinIO object store for the distributed file system.

Once the meta-information of the corresponding dataset to be ingested is provided and the files are validated and finally uploaded to the object store, then the Web interface interacts with the *data source connectors* to actually perform the data capturing process and make the data available to the ingestion pipelines via a Kafka Broker. As depicted in figure 6, the *data source connectors* can either retrieve the data from the private data lake (the MinIO Object Store), or connect to the corresponding OpenAPIs of the opt-in tools. As data are being captured from either sources, they are being internally transformed to a generic type, that is being forwarded to the corresponding target consumer. In the COMFORTage platform, the target consumer is a Kafka Broker. Once data arrives to a specific topic of the Kafka Broker, they can be now consumed by the following functions of the data ingestion pipelines.

7.1.1 Technical description

The **data source connectors** are implemented as a single java process that exposes a well-defined OpenAPI, and takes the responsibility to capture and retrieve data from various and heterogeneous data sources (using a variety of different **source connectors**), transform the data into an internal common structure and finally use a variety of different **converters** to eventually forward the initially captured data to a target destination. The following figure provides the high-level overview of its internal structure.

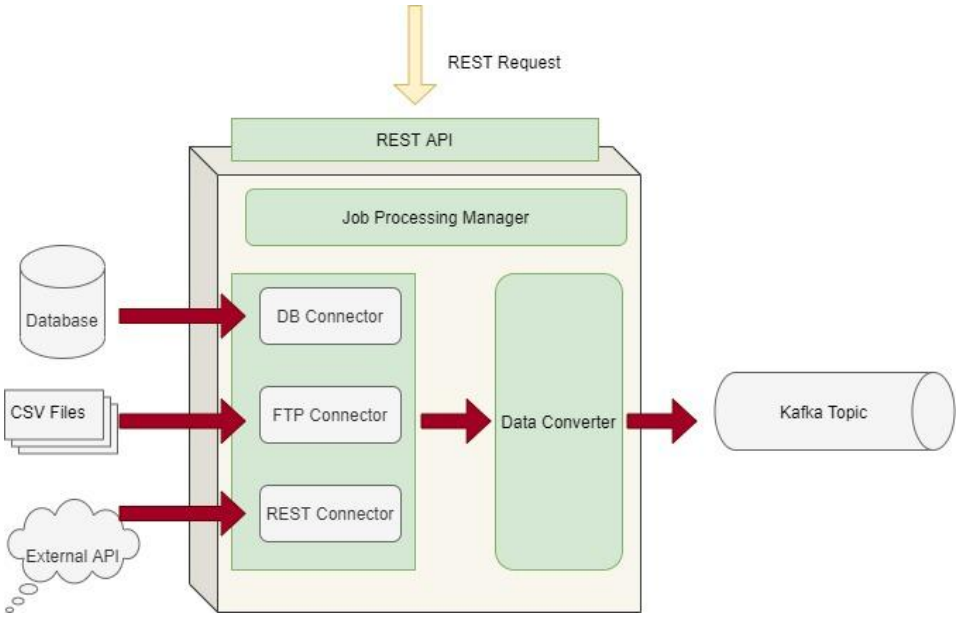


Figure 7: High-level overview of the Data Source Connectors

As it can be noticed from this figure, there is a top layer that handles the REST endpoint communications, the **service layer** that is responsible to implement the corresponding business logic and create the corresponding **jobs** to capture the data, and then, there are different types of connectors, each one of those is responsible to capture data from a specific source. Last but not least, a family of different **converters** are responsible to make this data available to the target consumer. A variety of different consumers are available (i.e. encrypted data via Avro Schema into a Confluent broker, a database, etc.), but in the scope of COMFORTage, we will rely on a single Kafka broker.

The **data source connectors** have been implemented in Java, using SpringBoot and the best practices for code factoring. As a fact, the following figure goes a step deeper, and provides more information regarding how the source code is packaged.

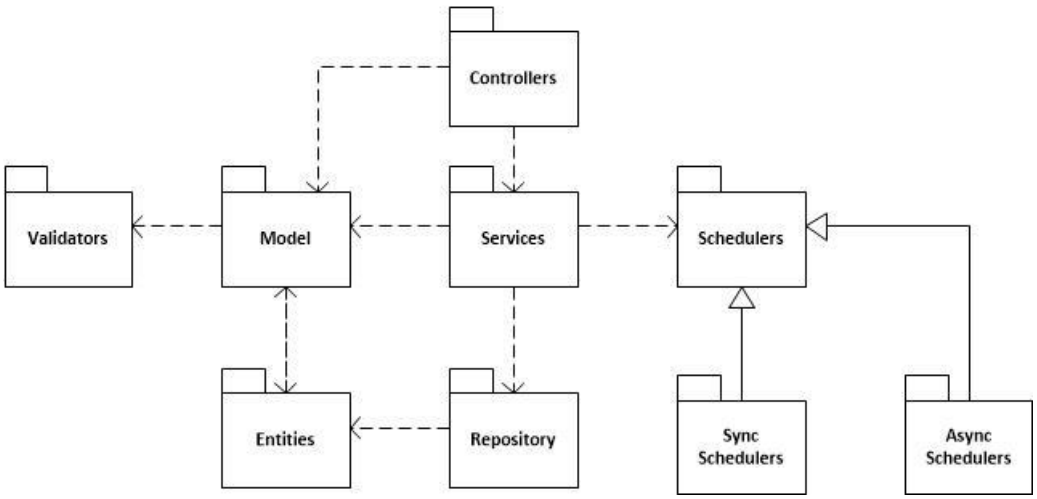


Figure 8: Data Source Connectors Component Diagram

From this diagram, we can distinguish the various packages of the source code. The **Controllers** take the responsibility to provide the OpenAPIs for the component interactions, while the *Services* contains the source code responsible for implementing the business logic of each of the OpenAPI interfaces. They both rely (request and respond) on messages that are defined under the *Model*

package, that in turn, uses the **Validators** package to perform any validation on the data that is being submitted into the web functions that lie under the **Controllers**. Examples of such validations might be non nullable fields, a correct ordering of submitted jobs, valid schema definitions etc. The purpose of the **Services** is twofold. From one hand, to perform the actual capturing of the data from the external sources (in the sense of creating specific **Jobs** to handle this), on the other hand, to persistently store all the information and statuses of these **Jobs**. For the latter case, they rely on the **Repository** package that takes the responsibility to persistently keep this information into a data store, and for the former case, they rely on the **Schedulers** package. The **Repository** makes use of the **Entities**, which in fact specifies the java persistent entities, as defined in the JPA specification. Both **Entities** and **Models** can be transformed back and forth, however **Models** are intended to be used as the **Data Transformation Objects (DTOs)** from within the overall application or while communicating with external sub-components, while the **Entities** are intended to be used for the direct communication of the application with the underlying data management system. Last but not least, the **Schedulers** take the responsibility to handle the corresponding **Jobs** that will capture the data from the corresponding sources and will convert and eventually make the data available to the corresponding target consumer. These **jobs** can be executed either in an asynchronous manner or can be scheduled to act in a periodic basis.

Regarding the **Jobs**, the following class diagram goes into more details of how they are implemented.

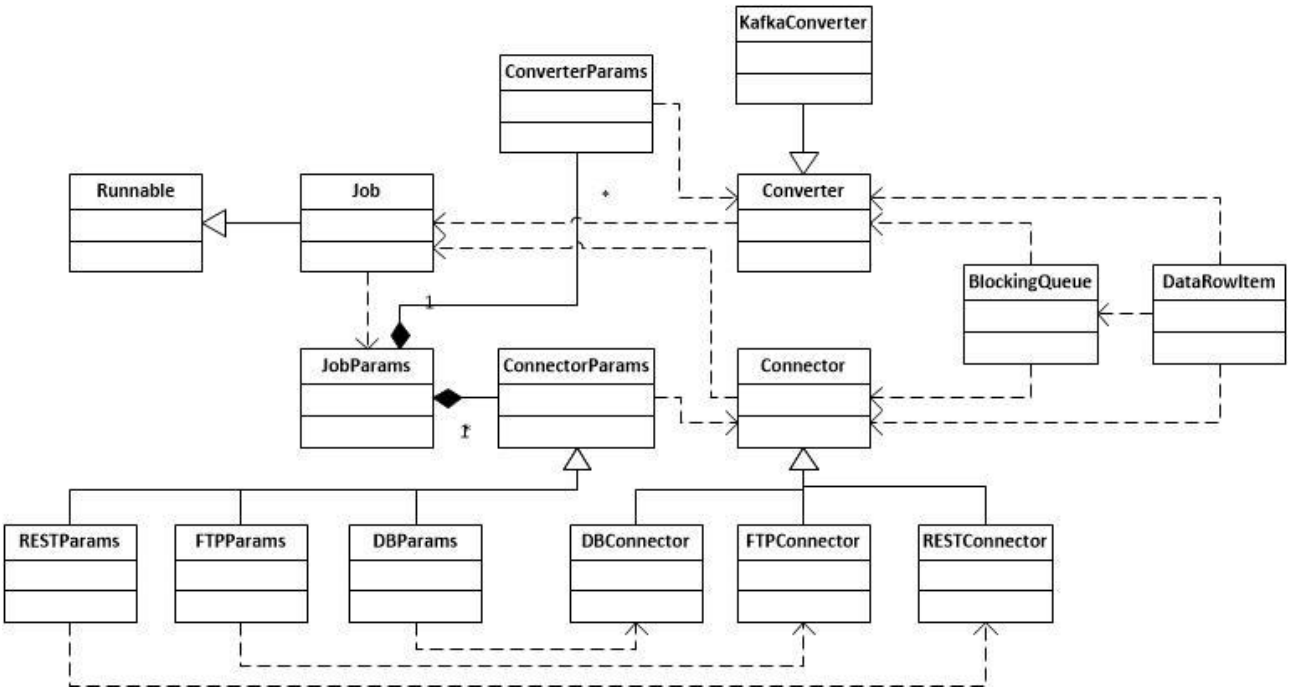


Figure 9: Data Connectors Job Class Diagram

From this figure, we can see that the *Job* abstraction contains a pair of a **connector** and a **converter**. Besides this, it also contains the relevant **JobParams**, which in turn, contain a pair of **connector** and **converter** parameters. These abstract parameters can be further materialized by the corresponding type of connectors or converters. For instance, an **FTPParam** would include the filepath of the file and the possible bucket of a distributed file system that uses FTP protocol to retrieve the data from, while the **RESTParam** will contain the IP, port and authentication mechanism of an external OpenAPI (provided by an opt-in tool) for the **RESTConnector** to establish the communication. Similarly, the **ConverterParams** provides the abstraction of all the implementation of the Converter, and more

precisely, the **KafkaConverter** will make use of the corresponding implementation that defines the IP, port and topic name of the target Kafka broker, for the converter to send the data to.

As mentioned, the **Job** instantiates the corresponding pair of connector and converter, according to their parameters. Once the instances are created, then the connector starts capturing data from the defined source. No matter the data source, each connector has the responsibility to transform the data source specific data items to the common **DataRowItem** structure, that is passed to an internal **BlockingQueue**, shared between the connector and the converter. The converter is listening to this queue, and once the data is available, it creates the corresponding messages to be sent to the target consumer. In our case, this is the target topic of a Kafka broker. Both the connector and the converter run on different threads, managed by the **Job** instance. In fact, the connector runs on a separate thread, while the converted runs on the initial thread that manages the execution of the **Job**. One thing to be highlighted here is that data is being sent to the target consumer into batches. This means, that the converter does not wait for all the data to become available, rather than creates messages of configurable size, each one of those containing data items of that particular size. By doing this, we can enable parallel processing of the **data ingestion pipelines**, as the functions defined there can work in parallel on small batches of data, rather than having to wait for the whole dataset to become available.

Another thing to be noted is that the **Job** abstraction itself implements the **Runnable** interface of Java. That means that each **Job** can be dispatched to corresponding **task schedulers** that will take the responsibility to actually run the submitted job. Task schedulers can either execute the job in an asynchronous manner, or schedule it to be performed in a periodic basis. That way, we can separate the concerns of implementing the business logic of the **Job** itself, with how this job can be later processed. Jobs takes the responsibility to connect to the external sources, capture data, convert them into our common structure and forward them to the kafka broker, while the classes under the **Schedulers** package are responsible to execute these jobs.

7.2 Technical specification

7.2.1 API documentation

The **Data Source Connectors** software artefact provides four different types of functionalities, each one of those allows the user to submit a new job, stop pending jobs, and get their current statuses. The four types can be summarized as follows:

- JobController: allows creating a single job for capturing data from an external data source, get the list of all submitted jobs (along with all their information like data started, status etc.) or stop a pending job.
- JobScheduledController: it creates a single job for capturing data that will be scheduled to perform in a periodic basis. It also allows retrieving a list of all scheduled jobs and stop already scheduled jobs.
- ComboJobController: it creates a **combo job**, which consists of a series of different individual jobs, with a concrete ordering. Individual jobs of these **combo jobs** will be executed one by one, according to the specified order. Again, it allows to retrieve the list of all combo jobs, and stop already pending ones.
- ComboJobScheduledController: it creates **combo jobs** that will be scheduled to perform in a periodic basis. It also allows retrieving a list of all scheduled combo jobs and stop already scheduled ones.

An example of job definition can be depicted in the following code snippet:

```
{
 "datasourceID": "Eginitio",
 "datasetID": "albion",
 "schema": {
  "type": "record",
  "name": "albion",
  "namespace": "eu.comfortage.eginitio",
  "fields": [
   {
    "name": "SEX",
    "type": [
     "double",
     "null"
    ]
   },
   {
    "name": "BIRTH_DAY",
    "type": [
     "int",
     "null"
    ]
   }
  ]
 },
 "schemaKey": {
  "type": "record",
  "name": "albionPK",
  "namespace": "eu.comfortage.eginitio",
  "fields": [
   {
    "name": "ID",
    "type": "string"
   },
   {
    "name": "VisitNuA",
    "type": [
     "double",
     "null"
    ]
   }
  ]
 },
 "connectorArguments": {
  "type": "FILE",
  "path": "/home/test//ALBION_DATASET_small.csv",
  "datePattern": "yyyy/MM/dd HH:mm:ss",
  "timePattern": "h:mm:ss a",
  "delimiter": ";",
  "nullString": "#NULL!",
  "skipFirst": true
 },
 "converterArguments": {
  "type": "PRINT",
  "datePattern": "yyyy-MM-dd HH:mm:ss z"
```

```
 }
}
```

In this code snippet, there can be noticed the several parameters required, such as the name of the data source provider, the name of the corresponding dataset, the schema (both for key and the values) of the datasets in Avro format, along with the specific parameters for the pair of connector and converter. In this example, we used the FILE type of connector, that requires the complete filepath of the csv file to be captured, the format of the timestamps, the delimiter etc, while we rely on the PRINT type of converter, that instead of forwarding the data items into a Kafka topic, it writes them to the console for testing purposes.

Similarly, a scheduled job example can be depicted in the following code snippet:

```
{
 "datasourceID": "Eginitio",
 "datasetID": "albion",
 "schema": { ...} ,
 "schemaKey": {},
 "connectorArguments": {},
 "converterArguments": {},
 "schedule": {
  "future": {
   "time": 5,
   "unit": "SECONDS"
  },
  "periodic": {
   "time": 10,
   "unit": "SECONDS"
  }
 }
}
```

In this code snippet, in addition to the previous arguments, we also have the *future* or *periodic* arguments, which configure the task scheduler to perform the job in the future, or periodically. The message could contain one of those, or both, but not none.

A combo job example can be depicted in the following code snippet:

```
{
 "name": "A combo job",
 "jobs": [
  {
   "order": 1,
   "job": { ...}
  },
  {
   "order": 2,
   "job": {...}
  }
 ]
}
```

Here, the combo job contains a list of jobs with a specific ordering, that will be scheduled to be executed sequentially, one after the other.

Finally, a scheduled combo job is a combo job, with the scheduled parameters, as the following code snippet depicts:

```
{
 "name": "A scheduled combo job",
 "jobs": [
  {
   "order": 1,
   "job": { }
  },
  {
   "order": 2,
   "job": {}
  }
 ],
  "schedule": {
  "future": {
   "time": 5,
   "unit": "SECONDS"
  },
  "periodic": {
   "time": 10,
   "unit": "SECONDS"
  }
 }
}
```

It is also important to highlight that the **data source connectors** software artefact complies with the OpenAPI specification, using Swagger 3. Therefore, the complete documentation of the API can be found there, as explained in the *user guide* section.

7.2.2 Technology stack and license information

The **Data Source Connectors** consist of two separate subsystems: the microservice layer that provides the overall functionality, along with an optional persistence layer.

The microservice has been implemented in Java 17, using SpringBoot 3.3.4. It relies on Maven for dependency management and build automation, while internally it makes use of the following dependencies:

- spring-boot-starter
- spring-boot-starter-web
- spring-boot-starter-webmvc-ui
- spring-boot-starter-data-jpa
- spring-boot-starter-validation
- spring-boot-starter-hateoas
- lombok
- org.json.json
- org.apache.avro.avro
- org.apache.kafka.connect-api
- org.apache.kafka.kaflaclients

- io.confulent.kafka-avro-serializer

Regarding the persistence layer, by default, it makes use of the in-memory H2 database, which is loaded in the context of the java process upon initialization. Therefore, no additional component needs to be created for the context of the overall solution. The default behaviour comes with the drawback that all information that is intended to be persistently stored, it is lost when the java process terminates. To truly enable the persistence layer, the user needs to use an additional MySQL datastore. In the *application.properties* file, there have been already defined the corresponding properties to enable the connectivity of the data source connectors with an instance of MySQL, so the user needs to uncomment these properties out.

All source code has been published under the MIT license model.

7.2.3 Code repository

The source code of the implementation is available under the COMFORTage's private code repository, in the following URL: https://git.comfortage.net/components/data-ingestion/data-ingestion-connectors

7.2.4 Improvements since previous version

The codebase of the data source connectors relies on the background technology brought to the COMFORTage platform from the iHelp project, and more precisely, its Data Capture Gateway. Even if both projects share common principles, there have been some significant advancements since its original version. These advancements consist of the foreground technology developed within the task T3.1 during the COMFORTage project. There can be summarized as follows:

- Storage layer added to persistently store the creation and statuses of the submitted jobs.
- Implementation of additional MinIO data connector
- Migration to Java 17
- Different java modules now packaged into a single SpringBoot application
- Code refactoring to separate different concerns into different packaged layers
- Distinct use of asynchronous and scheduled task executors
- Compatible with the OpenAPI specifications

7.2.5 Prerequisites and installation environment

The codebase of the **data source connectors** is written in Java 17, therefore the only prerequisites are the existence of Java 17 in the host machine, and maven 3.6 or greater, if the user wants to compile the software artefact from the source code.

Regarding resource requirements, this artefact has minimum requirements and can run in almost every recent commodity machine. However, it has been tested only in Linux distributions and thus Linux environments are highly recommended.

7.2.6 Installation Guide

The compilation process, after passing all the relevant unit tests, produces a **big fat jar** (a java archive that contains all required dependencies). The user needs to execute this java archive and the whole microservice will start, having the H2 in-memory database embedded.

In case the user needs to add a persistence layer, then a MySQL instance should be available. In this case, the following attributes need to be available in the application.properties file, before compilation (having the correct connection url, along with the relevant username and password):

```
## Properties for MySQL database ##
spring.datasource.url=jdbc:mysql://localhost:3306/test?createDatabaseIfNotExist=true
spring.datasource.username=root
```

```
spring.datasource.password=my-secret-pw
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
spring.jpa.hibernate.ddl-auto=update
```

In the next phase of the project, this software artefact will be also available as a docker image, so that the user can rely on docker, docker-compose, docker swarm or Kubernetes to facilitate the installation and deployment of this microservice.

7.2.7 User Guide

This software artefact interacts via OpenAPI specification, as described in a previous subsection. Therefore the complete documentation and user guide can be found there. Once the microservice has been started, the user can visit the following URL from the browser:

```
localhost:8081/swagger-ui/index.html
```

Once the page is loaded, the user can see the full documentation of this artefact, using the OpenAPI specification, as depicted in the following figure.

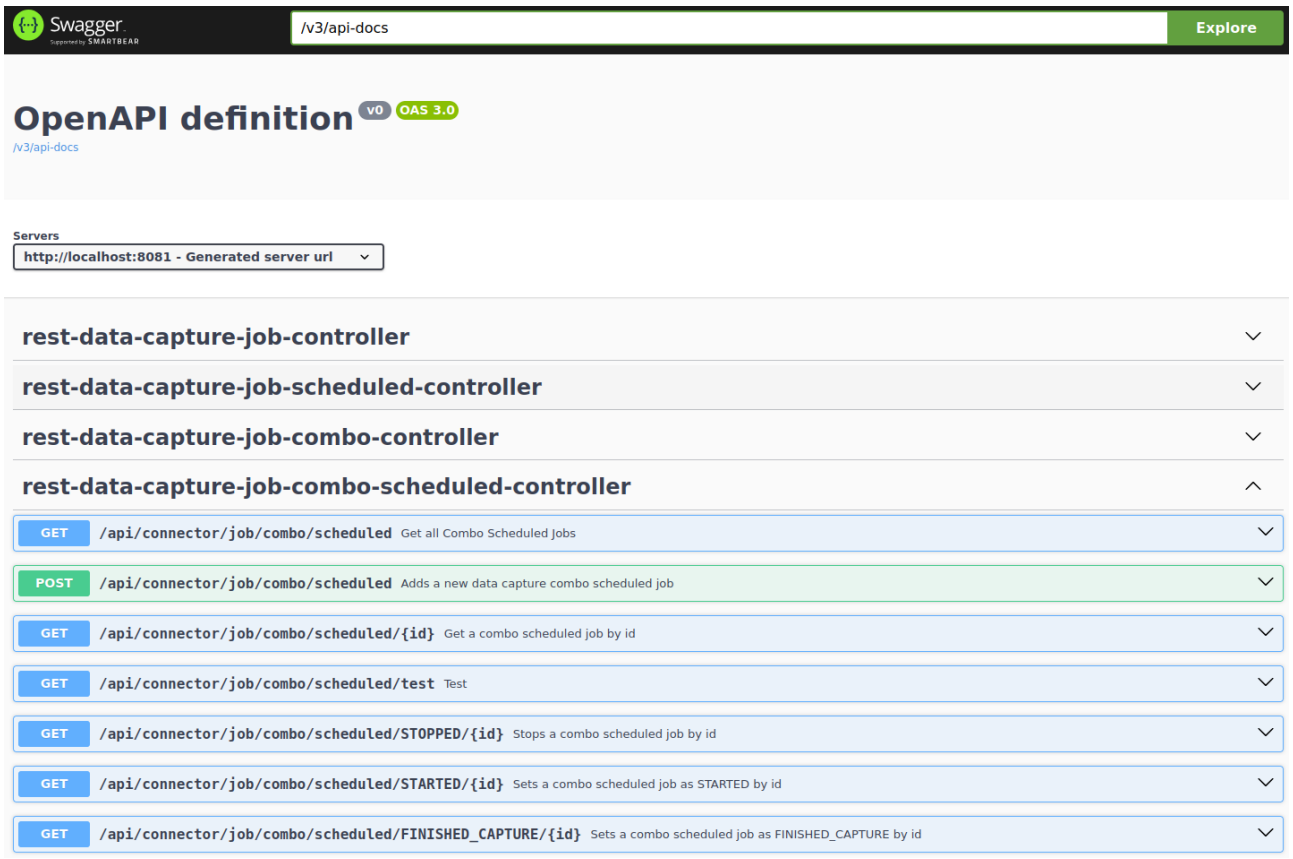


Figure 3: Data Source Connectors OpenAPI screenshot

There, we can see the four different types of controllers, already mentioned in a previous subsection, along with a family of corresponding web methods for each one of those. The OpenAPI specification also defines the exact format of the messages that need to be submitted, as depicted in the following screenshot.

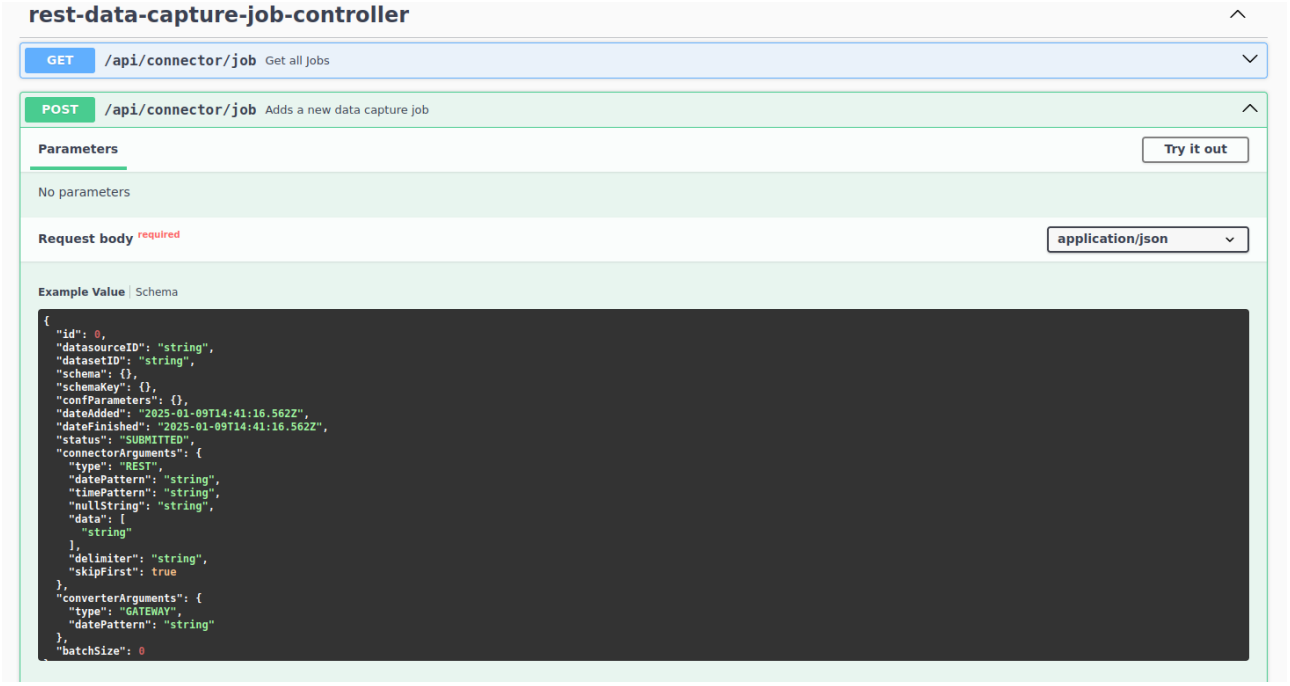


Figure 4: Example of a Create New Job Request

7.2.8 Considerations & future work

This section described the design and implementation of the **Data Source Connectors**, at this phase of the project. Regarding the future work, it can be summarised as follows:

- Dockerized version of the microservice
- Monitoring functionality to keep track of the progress and status of the submitted jobs
- Implementation of a novel **Kafka Connector** to enable the data capturing from a Kafka topic, most probably required by at least one of the opt-in tools
- Further unit tests to fully covered all provided functionality
- Additional OpenAPIs (controllers) to facilitate the integration with the WebUI of the overall **data ingestion pipelines**
- Generation of the Avro Schema definition, having the common dictionary as the input

7.3 Methodology for data collection and integration

7.3.1 Overview

The data mapping and integration process within the COMFORTage project follows a structured methodology to ensure consistency, interoperability, and usability of healthcare data. The technical partners, serving as data engineers, collaborate closely with pilot references to collect and structure the data necessary for AI-driven analysis and model computation. This methodology involves multiple phases, including data description, analysis, validation, and integration into a unified model. The process is iterative with multiple interactions between technical and pilot partners to better understand the data. This initial phase ensures that all relevant information also required for AI analysis is captured comprehensively.

7.3.2 Data collection and initial structuring

The data engineers gather structured data descriptions from pilot reference partners, detailing data entities, attributes, and their relationships. These descriptions encompass various types of healthcare data that organisations wish to store in the COMFORTage Integrated Knowledge Base.

During this phase, the collected datasets are examined to identify similarities, inconsistencies, ambiguities, and missing information. If gaps exist, additional details are requested from pilot partners, including the medical dictionaries, terminologies, ontologies used, attribute cardinalities, and constraints.

The mappings derived from individual pilot data descriptions are merged into a unified HHR model, consolidating various conceptual entities into standardised FHIR resources. If different pilots use similar concepts but represent them uniquely, a common conceptual class or subclass is created. A similar harmonisation process is applied to attributes and relationships within the datasets. This mapping approach standardises the representation of data, ensuring seamless integration within the COMFORTage IKB and AI-driven analytical tools. The data mapping process aims to define and structure the HHR model for the COMFORTage project, ensuring data interoperability and accessibility. As outlined in the methodology, multiple iterations of mapping and refinements may occur based on clarifications or the addition of new attributes.

As an illustrative example, a sample dataset provided by the ALBION partner is mapped against standardised FHIR elements. Each attribute is assigned its corresponding FHIR resource, ensuring that clinical data is represented in a structured and interoperable format. Through this ongoing process the ontologies describing the model behind COMFORTage HHR will be derived and finalized.

Table 1: ALBION Sample Dataset Description

CATEGORY	VARIABLE	LABEL	VALUES	MEASURE	FHIR	CODE
DEMOGRAPHICS	ID	PARTICIPANT'S ID		Scale	Patient.identifier	372039005
	SEX	PARTICIPANT'S SEX	1=male, 2=female, 999=missing	Nominal	Patient.gender	263495000
	BIRTH_DAY	BIRTH DAY		Scale	Patient.birthDate	
	BIRTH_MONTH	BIRTH MONTH		Scale	Patient.birthDate	
	BIRTH_YEAR	BIRTH YEAR		Scale	Patient.birthDate	184099003
CURRENT MEDICATION	MED1	MEDICATIONS: neuroleptics	0=NO, 1=YES, 999=MISSING	Nominal	Medication.code	372482001
	MED2	MEDICATIONS: memantine	0=NO, 1=YES, 999=MISSING	Nominal	Medication.code	406458800

8 IDSA Connector

8.1 Overview

Data spaces represent trusted and robust frameworks designed to manage the complete lifecycle of data. These frameworks encompass various critical elements, including diverse data models, metadata descriptors, and ontologies that provide semantic context and meaning. Additionally, data spaces integrate a suite of data services aimed at facilitating seamless access, processing, and analysis of data. The concept introduces a holistic approach to unify and manage all data sources within an organisation, regardless of their data model, format, or physical location.

The renewed interest in data spaces has been spurred by a global realization of the inefficiencies caused by isolated data practices. Many organisations function as "data silos" or "data islands," where data remains locked within departmental boundaries or isolated systems. This fragmentation results in significant barriers to data discovery, trusted sharing, and standardised exchange. Without effective procedures for ensuring interoperability, organisations face a waste of resources due to redundant and repetitive data-related activities.

In essence, data spaces share similarities with data marketplaces—online transactional platforms where data can be bought and sold. According to Snowflake, these marketplaces facilitate the exchange of data by providing a trusted environment for discovery, negotiation, and exchange. However, data spaces go beyond the marketplace concept by enabling fully decentralized infrastructures that promote peer-to-peer or federated interactions. This decentralization empowers participants—referred to as actors—to assume dual roles as both data providers and data consumers. Interactions within data spaces may involve negotiations and agreements, often settled for a specified monetary value or other mutually agreed terms.

Beyond data exchange, data spaces extend their utility by supporting the provision and consumption of data-driven services. This additional capability transforms them into comprehensive platforms that enable the development of next-generation applications. By combining secure data sharing, decentralized interaction, and service integration, data spaces present an advanced paradigm for addressing the challenges of data interoperability and collaboration in modern organisations.

The healthcare domain puts new challenges in the current data space ecosystem, trying to fulfil a diverse set of user and technical requirements, as identified by the work that is being conducted by the European Health Data Space association and mentioned in a previous sector. As most of these challenges are trying to be addressed by the Reference Architecture Model of the International Data Space Association (RAM of IDSA), in COMFORTage, we designed our integrated solution to be in compliance with this initiative.

8.2 Reference Architecture Model of the International Data Space Association

The Reference Architecture Model (RAM) of IDSA is a comprehensive framework that defines the principles, components, and processes required to establish secure and trusted data ecosystems. It

serves as a guide for organisations to create data spaces that facilitate data sharing, interoperability, and sovereignty while preserving data privacy and security.

It is built on a foundation of principles that guide its design and implementation: i) **data sovereignty**, whose purpose is to ensure that data owners retain control over their data, deciding who can access it, under what conditions, and for what purpose; ii) **interoperability**, which promotes standardisation and compatibility across systems, applications, and data formats to enable seamless data exchange; iii) **security and trust**, aiming to incorporate robust mechanisms to ensure data integrity, authenticity, and protection against unauthorised access; iv) **decentralization**, whose main objective is to advocates a distributed approach to data management, avoiding central control points and enhancing resilience; and finally, v) **ecosystem** orientation that facilitates collaboration among diverse participants, including data providers, consumers, and intermediaries.

The IDSA-RAM is structured into distinct layers that align with its components:

- Business Layer: Focuses on business ecosystems, governance, and economic interactions.
- Functional Layer: Specifies services, roles, and interactions within the data space.
- Process Layer: Describes workflows, use cases, and participant interactions.
- Information Layer: Covers data formats, metadata, and ontologies for semantic interoperability.
- System Layer: Defines the IT infrastructure, including connectors, clearing houses, and apps.
- Security Layer: Encompasses identity management, encryption, and trust mechanisms.

In this working paper, we focus on the business layer, which can be depicted in Figure 5.

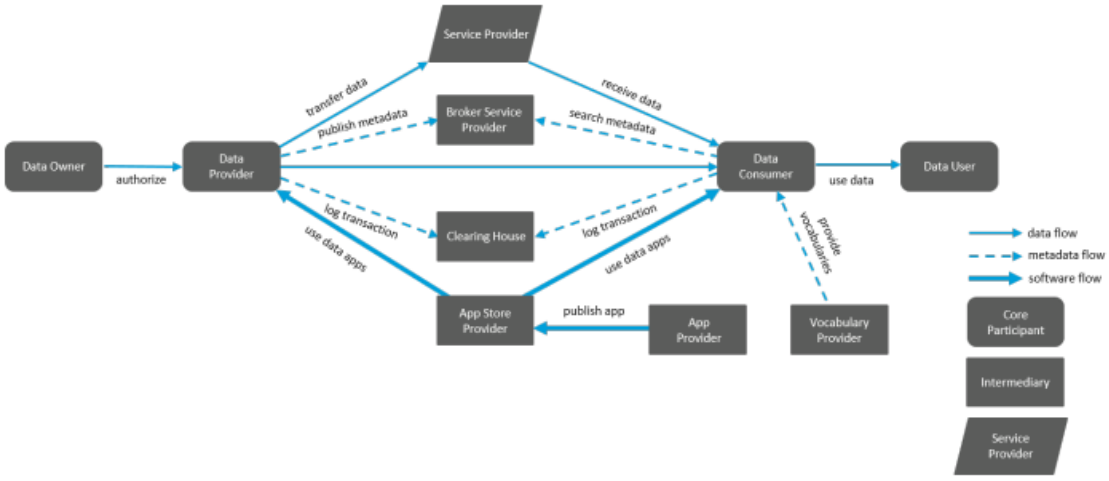


Figure 5: IDSA-RAM Business Layer

From this figure, we can distinguish different roles that participate and collaborate in the IDSA compliant data spaces. First and most important role is the one of the **data owner**. With the term **data owner**, the IDSA-RAM refers to the organisation that owns the actual data. It is important to highlight that the **data owner** is not always the **data provider** according to the IDSA terminology. In the COMFORTage project, the **data owner** can be a clinical organisation, a research center or opt-in tool that collects or generated primary or secondary data, and wants to make this data available to third parties and external actors through the concept of the EHDS, but does not have the technical

expertise or know-how to achieve this. Due to that, it exploits the benefits of the COMFORTage integrated solution.

The second actor according to the IDSA terminology is the **data provider**. The latter is the actor responsible to perform any data exchange on behalf of the **data owner** and other entities. The **data provider** uses software components that are compliant with the Reference Architecture Model of the International Data Spaces. Such a software component would be an IDSA certified **data connector**, which implements the mechanism necessary for this data exchange following the principles of interoperability and sovereignty. As a result, the COMFORTage integrated solution plays the role of the **data provider***,* which means that it provides the necessary tools to the **data owners** in order for the latter to make their data available via the EHDS. It is important to highlight that **data owner** and **data provider** may co-exist in the premise of the clinical organisation. This scenario is materialized when the clinical organisation deploys the COMFORTage integrated solution on its premises, thus it acts as both the **data owner** and **data provider***,* according to the IDSA terminology and specification.

Similarly, the **data user** and **data consumer** are the actors responsible to make use of data available through the IDSA compliant data spaces. The **data user** is the entity or the organisation that is interested in exploiting the data made available by **data owners** through the corresponding **data providers**. The **data consumer** provides the RAM-IDSA compliant software and service artefacts that are involved into the data exchange between the **data consumers** and **providers**. As in the previous case, **data users** and **data consumers** may co-exist in the same organisation or they could be materialized by two distinct actors.

Apart from these four main roles, there are others that can be categorized as **core** roles or **intermediates**. **Core roles** could be any **data app** compliant with the system architecture of the International Data Spaces and can be certified by a **Certification Body** towards increasing trust in these applications. They can either enhance the provided datasets by forcing FAIR or other cleaning and harmonisation attributes (therefore sitting on the **provider** side), or they can offer processing functionalities in the sense of AI predictions or modelling (therefore sitting on the **consumer** side). Other **intermediate** *roles* are the **broker service providers** that store and manage information about the data sources available in the International Data Spaces, **clearing houses***,* whose role is to provide clearing and settlement services for all financial and data exchange transactions, **vocabulary providers** responsible to manage and offer vocabularies (i.e., ontologies, reference data models, or metadata elements) that can be used to annotate and describe datasets and others.

In COMFORTage, we offer our Ageing-EHDS solution to be compliant with the Reference Architecture Model of the International Data Spaces and therefore, we do comply with the aforementioned business roles. However, we focus on the **data user** and **data provider** business roles, as all other roles are outside the scope of the project. As a result, our main purpose is to provide the IDSA compliant data connector that allows external users to make use of our available datasets.

8.3 Implementation

According to the latest report[1], there are currently more than 30 different implementations of IDSA compliant **data connectors**. These connectors should implement a list of different features or services, as depicted in the following figure.

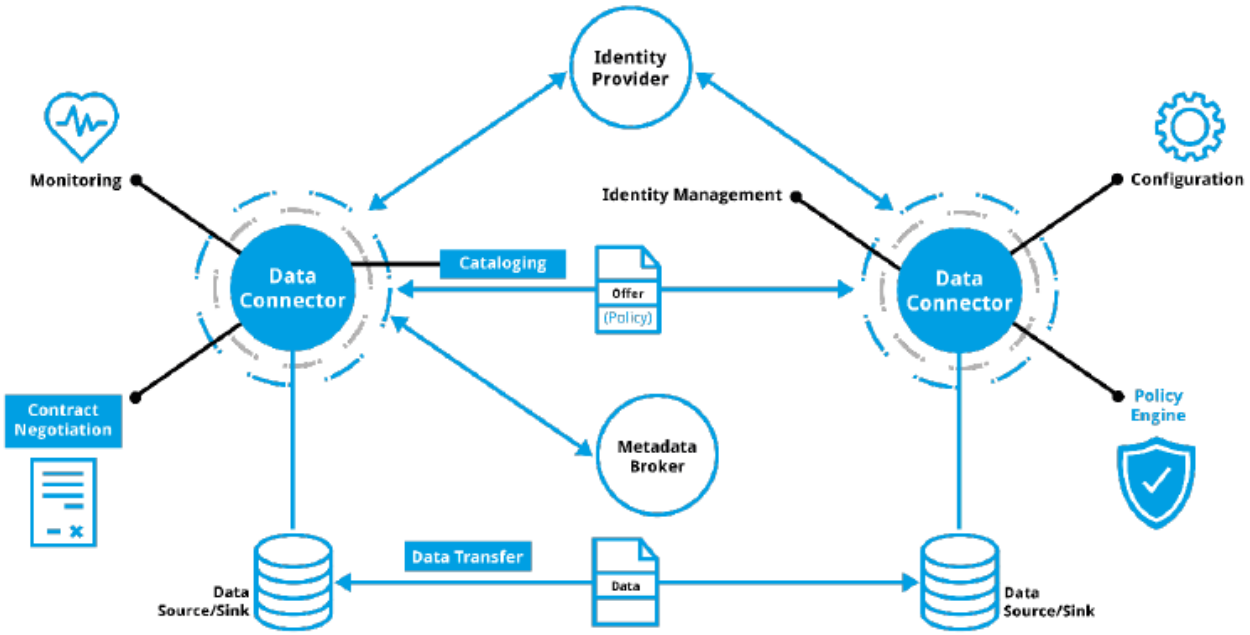


Figure 6: Data Exchange Services realized by an IDSA compliant data connector

More precisely, the **data connector** could act as both **data providers** or **data consumers**. In the case of **data provider**, they register available datasets that can be added as specific **data offerings**, listed in the **catalogue** of available **offerings**. This catalogue may or may not be further published into a **meta-data broker**, in order to facilitate their discoverability by external actors that are willing to make use of them. In the case of **data consumers**, they can explore the catalogue of the corresponding **data providers**, and then negotiate for a contract that will permit the consumers of actual make use of the dataset of interest. After the negotiation, once a contract is being officially agreed, it is accompanied by a rule set of **policies** that clearly define how the **consumer** is allowed to access the data. Having the contract with the set of **policies** agreed, the **data consumer** and **data providers** can now exchange or transfer data from the **source** (where the data resides in the **data owner**) to the **sink** *(*where the data should arrive in the **data user**).

In COMFORTage, we plan to rely at the first phase of the project to the open-source implementation of IDS-Configuration-Manager[2] for the implementation of the **data connectors**. This implementation provides all the functionality needed to define *a* **data source offering**, perform the negotiations with the data consumer, and finally offer this data source to be downloaded from the consumer. The following figure provides the overall architecture of this implementation.

[1] https://internationaldataspaces.org/data-connector-report/
[2] https://github.com/International-Data-Spaces-Association/IDS-ConfigurationManager

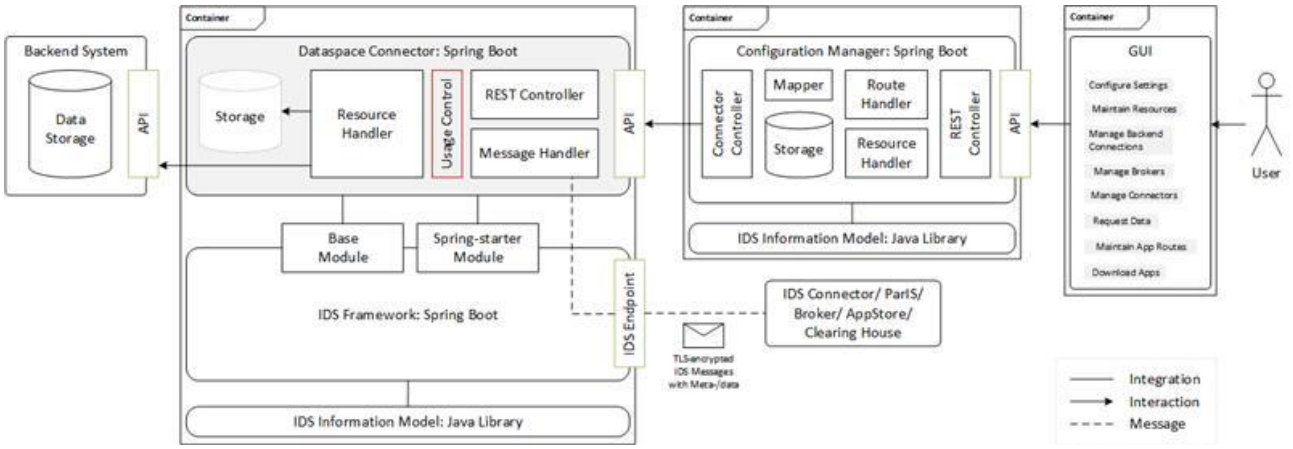


Figure 7: Data Connector overall architecture

From this figure, we can see the distinction between various components, depicted as different containers. Apart from the GUI interface, we can notice the **configuration manager** container, the **data space connector** container, and finally the **backend system**, that actually holds the datasets themselves into their corresponding data storage or data management system. With our approach, both the **configuration manager** and the **data space connector** container are provided by the open source implementation that we chose, while the COMFORTage integrated solution can be logically depicted as the **Backed System** from the figure above.

Translating this figure to the COMFORTage integrated solution, the role of the internal **Data Storage** is being materialized by the platform's **IKB**, while the corresponding API needed by the **data connector's resource handler** to interact with the underlying backend will be an additional microservice that will be provided by the platform. This microservice will play the role of the **bridge** between the COMFORTage platform's backend and the **data connector**. It is important to highlight at this point that the **data connector** requests from the **backend system API** to provide the available data sources via REST endpoints, and that is the main reason why we chose to put a microservice to implement this API and act as the **bridge** between the **data connector** and **IKB**, as the following figure depicts.

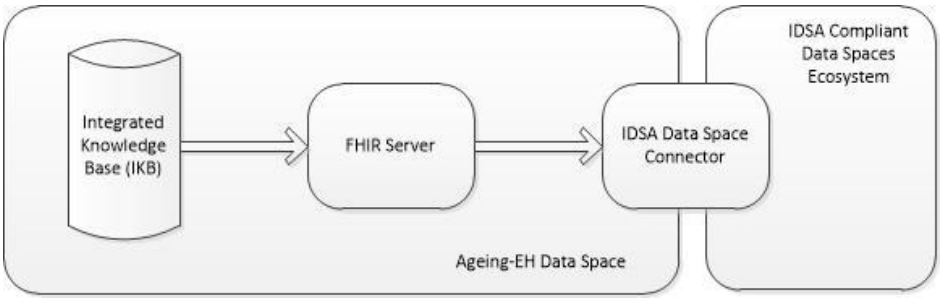


Figure 8: Ageing-EHDS integration with IDSA Data Spaces

This figure illustrates how the COMFORTage integrated solution can be part of the European Health Data Spaces, through its Ageing-EHDS solution. The IDSA Data Space Connector plays a crucial part in the design, as it allows our Ageing-EHDS solution to be part of the IDSA Compliant Data Space Ecosystem, as described above. In that sense, the Ageing-EHDS plays the role of the **data provider**, according to the IDSA terminology, which makes use of its internal **IKB** for its underlying backend data management system. The integration with the **data connector**, as highlighted in both Figure 7 and Figure 8 is being provided by the novel **FHIR Server** microservice. The purpose of this planned

microservice is to firstly deliver data from the IKB via REST endpoints, as required by the **data connector**, and also to ensure interoperability of the data. In fact, the FHIR microservice will be compliant with the HAPI-FHIR specification, for querying and consuming electronic health records. It specifies how the REST API should be exposed in order for the data user to be able to retrieve data from in a standardised manner. Last but not least, the REST endpoints will serve standard FHIR resources in order to foster the re-usability of data among different tools and services. It is important to highlight also that the IKB internally complies with the HHR model of the COMFORTage project, which is the internal conceptual domain model of the project, which complies with the FHIR standard. More information about the implementation of the FHIR Server microservice will be given in the second version of this report.

8.4 Cross-sector potential

COMFORTage's data connector implementation uses the **open-source IDS-Configuration-Manager**, integrating with:

- **Metadata brokers** that facilitate discovery of available datasets under specific contractual terms.
- **Clearing houses** that notarise transactions for regulatory compliance and later audits.
- **OpenAPI-described endpoints**, making it straightforward for external systems (hospital IT, national health platforms, or even smart city data hubs) to integrate.

While initially focused on dementia and frailty datasets, this architecture **is inherently cross-sector capable**, supporting future integrations with:

- Smart home environmental sensors feeding into frailty risk scores.
- Urban mobility datasets exploring links between activity patterns and cognitive decline.
- Social care registries to identify loneliness indicators.

9 Innovation Highlights and Extended Lessons Learned

9.1 Highlights: 5 reasons why COMFORTage stands out

1. **First large-scale use of IDSA RAM in dementia & frailty health contexts**, demonstrating how industrial data governance models can secure sensitive health data.

2. **IAM with multi-realm architecture**, providing true logical isolation of different pilots and studies, vital when balancing local governance (e.g., specific regional GDPR interpretations) with shared analytics.

3.**Consent management deeply embedded in operational tokens (JWTs)**, moving compliance from static paperwork to dynamic, enforceable programmatic checks.

4. **Holistic data integration**, uniting smart home sensors, genomics, and clinical EHRs into a coherent analytical ecosystem.

5. **EHDS alignment by design**, minimising future retrofit costs and regulatory hurdles.

9.2 Lessons learned: technical, organisational, ethical

"Interoperability without governance is just technical debt deferred."

Building cross-border data ecosystems demands not just technical standards, but also shared governance, mutual trust certifications, and legal harmonisation.

- **Dynamic consent is indispensable.** Patients often change preferences over time, especially in long studies. Embedding consent into IAM tokens allows immediate enforcement without massive manual audits.
- **PBAC was more complex to implement than anticipated.** While RBAC is simple, policy-based controls required careful modelling of data sensitivity tags, processing purposes, and time constraints.
- **Privacy engineering upfront saves costs.** Architecting IAM and data connectors to enforce GDPR principles by design avoided expensive post-hoc compliance rewrites.
- **Living Labs revealed digital literacy challenges.** Some older adults were hesitant to engage with digital dashboards, requiring tailored UX designs and community workshops to build trust.

These lessons are invaluable not just for COMFORTage, but for any future EHDS-aligned initiative.

10 Outlook: Roadmap to a fully operational EHDS ecosystem

10.1 Immediate technical roadmap

As the project advances toward its next milestones, key technical focuses include:

- **Deploying row- and column-level masking in IAM**, enabling disclosures such as sharing cardiovascular markers for a study while hiding cognitive test scores.
- **Launching the production-grade FHIR microservice**, facilitating interoperability with hospital EHR systems, national platforms, or cross-border care initiatives.
- **Extending ontology mappings** to include new biomarkers and social determinants, crucial for holistic predictive modelling.

10.2 Strategic positioning for broader European impact

By building such close alignment with EHDS and IDSA frameworks now, COMFORTage sets itself up to:

- Serve as a **reference architecture** for other Horizon Europe or Digital Europe health data projects. While initially focused on dementia and frailty datasets, this architecture **is inherently cross-sector capable**, supporting future integrations with e.g.:

  - Smart home environmental sensors feeding into frailty risk scores.
  - Urban mobility datasets exploring links between activity patterns and cognitive decline.
  - Social care registries to identify loneliness indicators.

- Integrate smoothly with future Health Data Access Bodies for streamlined secondary use authorisations.
- Expand into multi-sector collaborations, connecting ageing health data with transport, housing, or urban planning datasets to address broader quality-of-life determinants.

11 Conclusion

The COMFORTage project exemplifies how advanced technological architectures, rigorous regulatory alignment, and participatory co-design can converge to build the secure, scalable, patient-controlled health data ecosystems envisioned by the European Health Data Space.

By embedding IDSA principles of data sovereignty, deploying IAM systems that make consent a living, enforceable attribute, and planning for SPE-based federated analytics, COMFORTage not only addresses today's pressing dementia and frailty challenges but also lays the foundation for a future in which health data can truly serve individuals and society without compromising privacy or trust.

As Europe continues to age, infrastructures like COMFORTage provide a clear pathway for how innovation, governance, and citizen empowerment can together reshape the landscape of health, dignity, and quality of life across the continent.

Funded by
the European Union