CrowdHEALTH

Collective Wisdom Driving Public Health Policies

Del. no. – D 3.5 Data Sources & Gateways: Design & Open Specification v.1

Project Deliverable

This project has received funding from the European Union's Horizon 2020 Programme (H2020-SC1-2016-CNECT) under Grant Agreement No. 727560

D 3.5 Data Sources & Gateways: Design & Open Specification v.1

Work Package:		WP3
Due Date:		31/10/2017
Submission Date:		31/10/2017
Start Date of Project:		01/03/2017
Duration of Project:		36 Months
Partner Responsible of Deliverable:		SiLo
Version:		1.1
Status:	☒ Final ☐ Draft ☐ Ready for internal Review ☐ Task Leader Accepted ☐ WP leader accepted ☐ Project Coordinator accepted	
Author name(s):	Konstantinos Perakis (SiLo)	Dimitris Miltiadou (SiLo)
	Antonio De Nigro (ENG)	Francesco Torelli (ENG)
	Salvador Tortajada Velert (HULAFE)	
Reviewer(s):	Vegard Engen (IT Innovation)	Tomas Pariente Lobo (ATOS)
Nature:	☒ R – Report ☐ D – Demonstrator	
Dissemination level:	☒ PU – Public ☐ CO – Confidential ☐ RE – Restricted	

REVISION HISTORY			
Version	Date	Author(s)	Changes made
0.1	18/09/2017	Konstantinos Perakis (SiLo)	Draft – Index
0.2	22/09/2017	Konstantinos Perakis & Dimitris Miltiadou (SiLo)	Contributions to section 2
0.3	29/09/2017	Konstantinos Perakis & Dimitris Miltiadou (SiLo) Antonio De Nigro (ENG) Francesco Torelli (ENG) Salvador Tortajada Velert (HULAFE)	Contributions to section 2
0.4	02/10/2017	Konstantinos Perakis SiLo) Salvador Tortajada Velert (HULAFE)	Consolidation of Annex A
0.5	06/10/2017	Konstantinos Perakis & Dimitris Miltiadou (SiLo)	Contributions to section 3
0.6	13/10/2017	Konstantinos Perakis & Dimitris Miltiadou (SiLo)	Contributions to section 3
0.7	20/10/2017	Konstantinos Perakis (SiLo)	Conclusions
0.8	24/10/2017	Konstantinos Perakis (SiLo)	Review Ready Version
0.8_ATOS	27/10/2017	Tomas Pariente Lobo (ATOS)	ATOS Internal Review
0.8_IT-INN	27/10/2017	Vegard Engen (IT Innovation)	IT-INN Internal Review
1.0	30/10/2017	Konstantinos Perakis (SiLo)	Final version
1.1	08/11/2017	ATOS	Quality control and submission to EC

List of acronyms

API	Application Programming Interface
BIO	BIOASSIST SA
CRA	Care Across Ltd
DB	Database
EC	European Commission
ENG	ENGINEERING – INGEGNERIA INFORMATICA SPA
FHIR	Fast Healthcare Interoperability Resources
HHR	Holistic Health Records
HL7	Health Level 7, a set of international standards for transfer of clinical and administrative data between software applications.
HTTP	HyperText Transfer Protocol
HULAFE	FUNDACION PARA LA INVESTIGACION DEL HOSPITAL UNIVERSITARIO LA FE DE LA COMUNIDAD VALENCIANA
JSON	JavaScript Object Notation
KI	KAROLINSKA INSTITUTET
PHR	Personal Health Record
REST	Representational State Transfer
SiLO	SINGULARLOGIC CYPRUS LTD
SQL	Structured Query Language
TBD	To Be Defined
URI	Uniform Resource Identifier
WP	Work Package
XML	eXtensible Markup Language

Contents

List of Figures

List of Tables

Executive Summary

The scope of the current deliverable is to outline the architecture and design of the data sources and gateways framework. This framework aims at enabling the acquisition of multimodal data from various data sources and various data source providers, solving current connectivity and communication issues. Towards this end, the current deliverable documents the architecture and design of the data gateway component and the envisioned abstracted API for all information sources handled within the context of the CrowdHEALTH project. It documents the initial version of the component design, as this has been formulated building upon the requirements collected from the pilot partners of the project, who also act as data providers, while it also provides the initial version of the component specifications. It should be noted at this point that the design and specifications of the data gateway component are subject to changes that will be documented in the second version of the current deliverable. More specifically, the current version will guide the implementation of the first version of the Data Gateway component software prototype, which in turn will be evaluated within the context of the project and enriched in order to meet additional end user requirements or just be refined in its second and final version. It should also be noted that the actual connection details to specific data source providers will not be disclosed in the public deliverables, since this information is considered private by the data providers.

1. Introduction

The scope of the current deliverable is to outline the architecture and design of the Data Sources and Gateways framework. This framework aims at enabling the acquisition of multimodal data from various data sources and various data source providers, solving current connectivity and communication issues. Towards this end, the current deliverable documents the architecture and design of the Data Sources and Gateways component and the envisioned abstracted API for all information sources handled within the context of the CrowdHEALTH project. It documents the initial version of the component design, as this has been formulated building upon the requirements collected from the pilot partners of the project, who also act as data providers, while it also provides the initial version of the component specifications.

The deliverable is structured as follows:

1. The first chapter introduces the deliverable and explains its scope and positioning in the project, along with its relation with other deliverables.
2. The second chapter outlines the component design, providing information regarding the component scope, and the initial design of the component, illustrating its relation to the other components of the overall architecture.
3. Chapter 3 undertakes the analysis of the specification of the Data Sources and Gateways component, documenting the specification of the internal subcomponents of the Data Sources and Gateways component which will drive its implementation.
4. Chapter 4 concludes the current deliverable and discusses future work.

Deliverable D3.5 is the first version of the report of the Data Sources and Gateways component. The current deliverable undertakes the documentation of the efforts undertaken within the context of Task 3.2 – Data Sources and Gateways, towards defining an abstracted and unified API so as to enable the acquisition of multimodal data from various sources and providers. Deliverable D3.5 receives as input the requirements analysis (Deliverable D2.1 – State of the Art and Requirements Analysis v.1.0) along with the Reference Architecture included in Deliverable D2.4 – Conceptual Model and Reference Architecture v.1.0. This deliverable in turn provides information to D3.19 undertaking the documentation of the design and specification of the data cleaning component (as part of the reliable information provision in healthcare approach) as well as to D3.9 undertaking the documentation of the advanced interoperability techniques, specifying the interconnection with the internal components of the architecture, with which the Data Sources and Gateways component will exchange information.

It should be noted at this point that the design and specifications of the Data Sources and Gateways component are subject to changes that will be documented in the second version of the current deliverable. More specifically, the current version will guide the implementation of the first version of the Data Sources and Gateways component software prototype, which in

turn will be evaluated within the context of the project and enriched in order to meet additional end user requirements or just be refined in its second and final version. It should also be noted that the actual connection details to specific data source providers will not be disclosed in the public deliverables, since this information is considered private by the data providers.

2. Data Sources and Gateways Component Design

2.1. Data Sources and Gateways Component Scope

The scope of Data Sources and Gateways component is to deliver the software tool supporting the process of acquiring multimodal data from various sources and various providers, which may be described in different and often inconsistent formats. The Data Sources and Gateways component aims to provide an abstracted and unified API which will support the acquisition of information from sources including healthcare organisations, sensors, laboratories, mobile applications and more. The delivery of the Data Sources and Gateways component will facilitate the resolution of the connectivity and communication challenges with such information sources, ensuring – through the interaction with the rest of the internal components of the CrowdHEALTH architecture – the integration of the syntactically harmonised - in terms of following one common specification format - as well as cleaned - in terms of being validated against a set of rules while having erroneous data corrected and missing data handled - information into the Holistic Health Records, constituting the final aspired outcome of the CrowdHEALTH platform.

As per the requirements documented in deliverable D2.1 of the project (Kyriazis et al., 2017), the Data Sources and Gateways component should:

1. Facilitate the connection to an appropriately specified (SQL or No-SQL) Database, for the retrieval of the information (Technical Requirement TL-FUNC-3221);
2. Facilitate the connection to an appropriately specified API, for the retrieval of the information (Technical Requirement TL-FUNC-3222);
3. Facilitate the parsing of files (e.g. excel or csv files, for the retrieval of the information) (Technical Requirement TL-FUNC-3223);
4. Provide access to a configuration service, facilitating configuration of the connection parameters per connection type and source (Technical Requirement TL-FUNC-3224);
5. Support pulling data from external data sources (e.g. through REST APIs) per predefined time intervals (Technical Requirement TL-FUNC-3225);
6. Support data from external data sources being pushed to the platform per predefined intervals (Technical Requirement TL-FUNC-3226);
7. Support token-based authentication with the data sources to safeguard data integrity and non-repudiation (Technical Requirement TL-FUNC-3227);
8. Support username and password-based authentication with the data sources to safeguard data integrity and non-repudiation (Technical Requirement TL-FUNC-3228);
9. Facilitate the standardised connection to other internal components of the CrowdHEALTH platform, such as the Data Cleaner, the Data Converter, etc (Technical Requirement TL-FUNC-3229);
10. Facilitate the connection to unknown, plug ‘n play sources, mapping them to already known sources in order to identify the information types made available (Technical Requirement TL-FUNC-32210);

11. Facilitate the interpretation of the information acquired from plug ‘n play sources connected, mapping them to already known sources in order to identify the information types made available (Technical Requirement TL-FUNC-32211);

The initial version of the Data Sources and Gateways component aims to provide an abstracted layer which will facilitate the integration of the different interface implementations of the sources provided by the project use case partners comprising the data providers, so that they could be handled uniformly from the CrowdHEALTH platform. As per deliverable D2.1, the first version of the Data Sources and Gateways component should support requirements #1 – #9 from the aforementioned bullet-list (corresponding to technical requirements TL-FUNC-3221 to TL-FUNC-3229).

Further, the second version of the Data Sources and Gateways component aims to provide an even greater level of abstraction, supporting the integration of unknown sources, connecting to the same exposed API and integrating additional information to the developed Holistic Health Records, supporting requirements #10 – #11 from the aforementioned bullet-list (corresponding to technical requirements TL-FUNC-32210 to TL-FUNC-32211).

2.2. Data Sources and Gateways Component Description

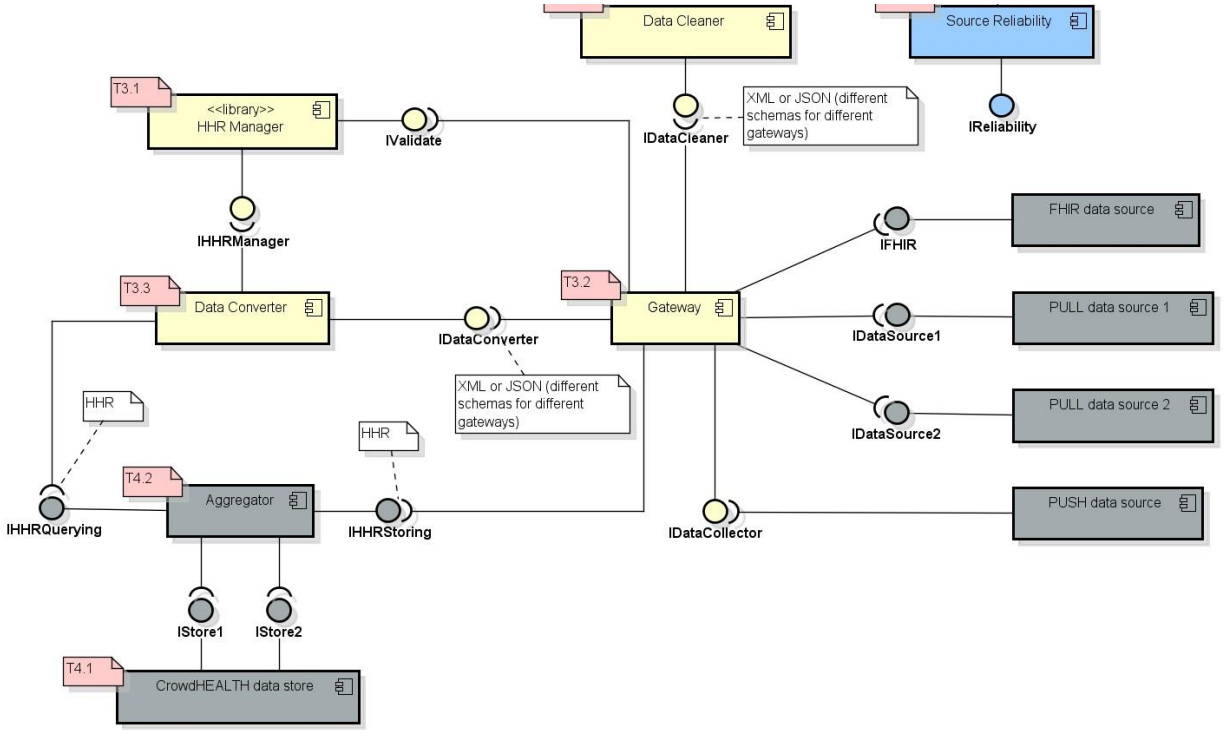

Figure 2-1. Data Sources and Gateways Architectural Positioning

Figure 2-1 provides the graphical representation of the positioning of the CrowdHEALTH Data Sources and Gateways component within the holistic project approach. As seen through this

schematic, the component exposes one generic interface to the rest of the platform for the acquisition of the information from external data sources (namely IDataCollector as per the schematic), while interfaces are also exposed to it from other internal – to the CrowdHEALTH architecture – components (namely IDataCleaner, IHHRStoring, and IDataConverter as per the schematic) so that aggregated information can be in turn forwarded to these components.

Figure 2-2 provides the graphical representation of the Data Sources and Gateways component, highlighting its internal sub components, as well as its interfaces with other internal components of the CrowdHEALTH platform.

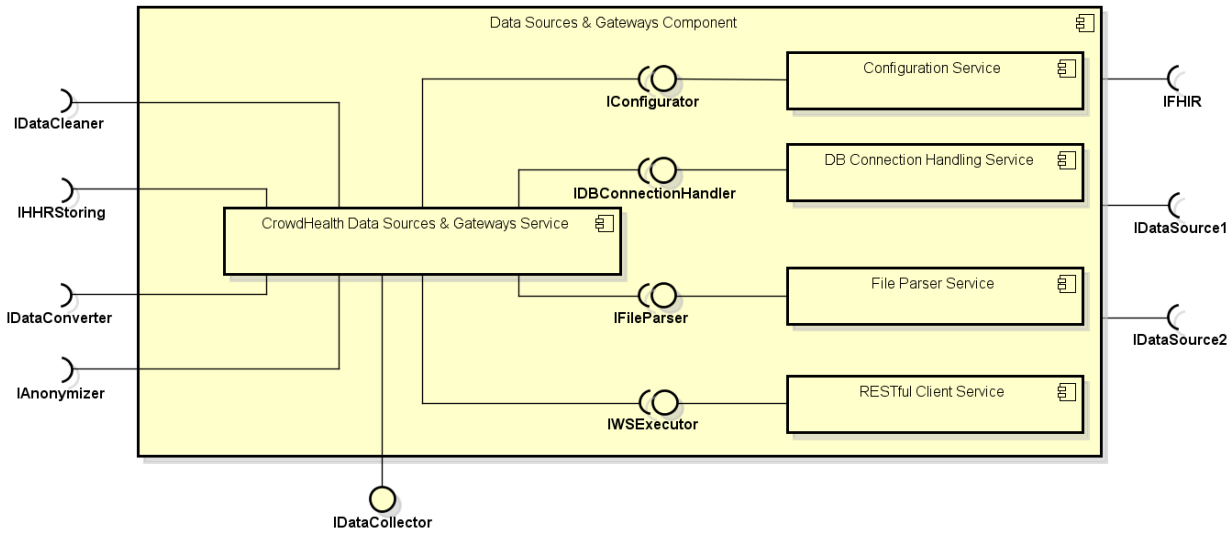

Figure 2-2. Data Sources and Gateways Component Design

The specific schematic can be considered to comprise of four main stripes:

- The external right stripe (including the IFHIR, the IDataSource1 and the IDataSource2 interfaces) represents interfaces exposed to the Data Sources and Gateways Service by the data providers and their description is thus out of the scope of the current deliverable.
- The internal right stripe (including the Configuration Service, the DB Connection Handling Service, the File Parser Service, the RESTful Client Service and their corresponding interfaces) represents the internal services and interfaces of the system. These services are transparent not only to the CrowdHEALTH platform users, but also to the rest of the CrowdHEALTH platform components as well. Each of the four internal services supported by the CrowdHEALTH Data Sources and Gateways component exposes one internal interface to the main CrowdHEALTH Data Sources and Gateways Service, which can be considered as the orchestrator of the various services. The four internal services are described in the paragraphs to follow, and more specifically in sections 2.2.2 to 2.2.5.

- The central stripe (including the CrowdHEALTH Data Sources & Gateways Service and the IDataCollector interface) represents the central CrowdHEALTH Data Sources and Gateways Service, responsible for handling all incoming and outgoing traffic on the CrowdHEALTH platform, and the connection options to it. The CrowdHEALTH Data Sources and Gateways Service is responsible for mediating between the internal components (configuration service, DB Connection Handling Service etc.) and for exposing the IDataCollector interface so as to facilitate information retrieval from the data providers. The CrowdHEALTH Data Sources and Gateways Service is described in more detail in section 2.2.1.
- The external left stripe (including the IDataCleaner, the IHHRStoring, the IDataConverter and the IAnonymizer interface) represents the interfaces that are external to the CrowdHEALTH Data Sources and Gateways Components, as per the holistic approach illustrated in Figure 2-1, which will not be described in the current deliverable since they will be exposed by other components to be discussed in other deliverables (i.e. the Data Cleaner component to be developed in the context of T3.5 exposes the IDataCleaner interface, the Data Converter to be developed in the context of T3.3 exposes the IDataConverter interface etc.).

As illustrated in the Figure 2-2, the CrowdHEALTH Data Sources and Gateways Service comprises of four main services:

1. The Configuration Service
2. The DB Connection Handling Service
3. The File Parsing Service
4. The RESTful Client Service

These four services were designed based upon the specifications of the project pilots. The filled in template that was used for the collection of the information from the pilots is included in Annex A of the current document.

The aspired information flow and the procedures and processes supported by the CrowdHEALTH Data Sources and Gateways component are illustrated in Figure 2-3 (example of pulling information from an API exposed by a data provider) and elaborated in the forthcoming paragraphs.

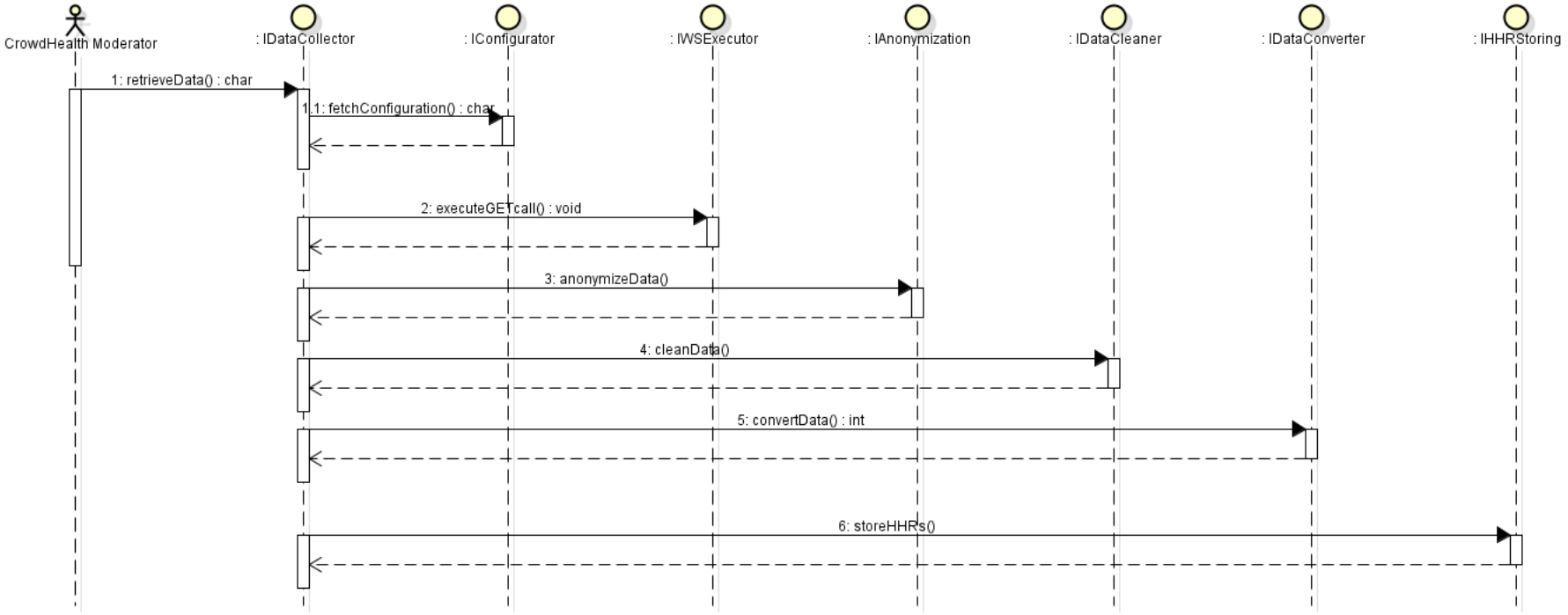

Figure 2-3. Data Sources and Gateways Component Sequence Diagram

As per Figure 2-3, the Data Collector interface (IDataCollector), which is the only interface exposed to the rest of the CrowdHEALTH platform by the Data Sources and Gateways component, is responsible for retrieving (in the scenario described, pulling) information from a specific data source provider, either upon trigger, or per pre-specified time intervals. Once required, the IDataCollector interface requests the configuration file (containing all connection details) of the specific information source in order to connect to it and start the information retrieval. The internal IConfigurator interface sends the request to the Configuration Service, and the Configuration Service replies with the appropriate configuration file. Upon receipt of the connection configuration file, the CrowdHEALTH Data Sources and Gateways Service through the IDataCollector interface initiates information retrieval, connecting (for example) to the external API provided by the data providers, invoking the corresponding internal RESTful Client Service through the internal IWSExecutor.

After information has been retrieved, it optionally needs to be further anonymised, on top of the at-source anonymization that has already taken place within the context of the CrowdHEALTH data pre-processing. Thus, the CrowdHEALTH Data Sources and Gateways Service triggers the interface exposed by the Anonymization component implementing the CrowdHEALTH anonymization approach (IAnonymizer as per Figure 2-2 and

This project has received funding from the European Union's Horizon 2020 Programme (H2020-SC1-2016-CNECT) under Grant Agreement No. 727560

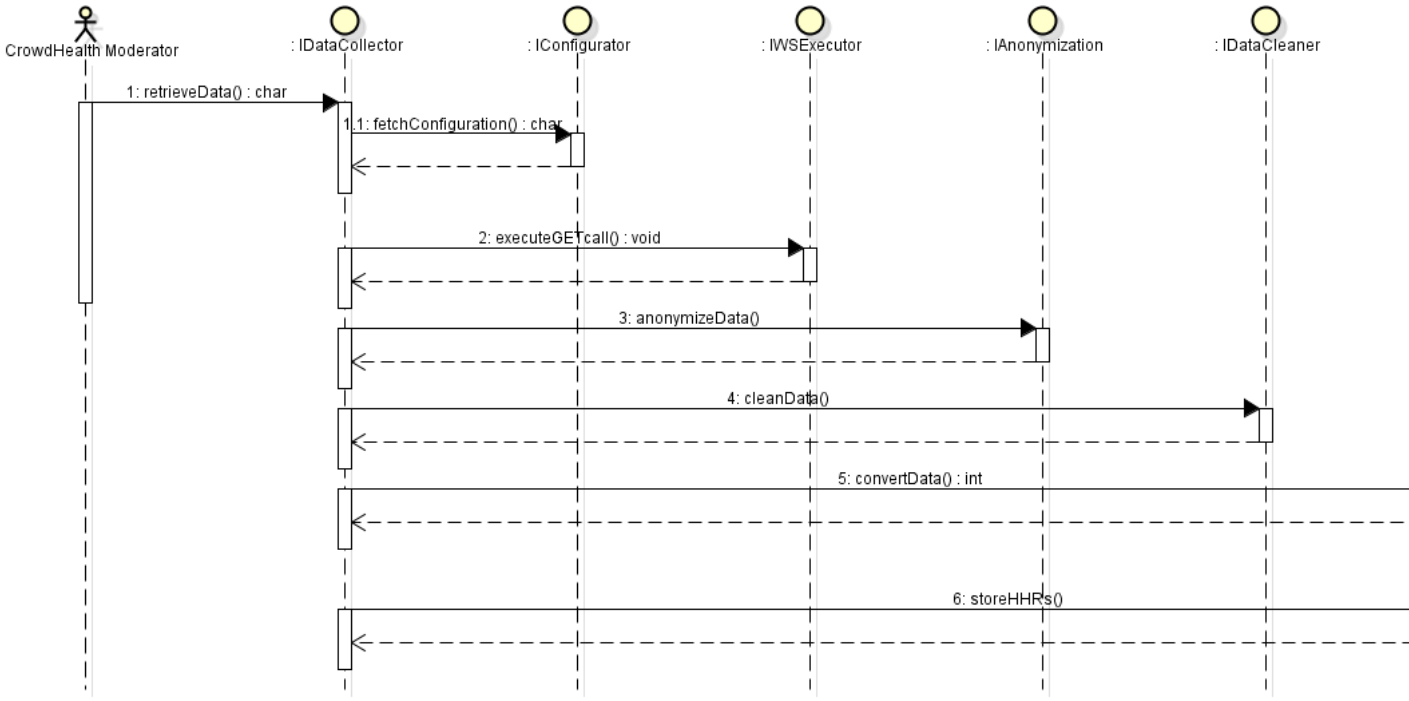

Figure 2-3), which in turn proceeds with the actual anonymization of the collected data and returns the anonymised data to the CrowdHEALTH Data Sources and Gateways Service.

Upon anonymization (if required), the CrowdHEALTH Data Sources and Gateways Service triggers the interface exposed by the Data Cleansing component implementing the CrowdHEALTH data cleaning approach (IDataCleaner as per Figure 2-2 and

This project has received funding from the European Union's Horizon 2020 Programme (H2020-SC1-2016-CNECT) under Grant Agreement No. 727560

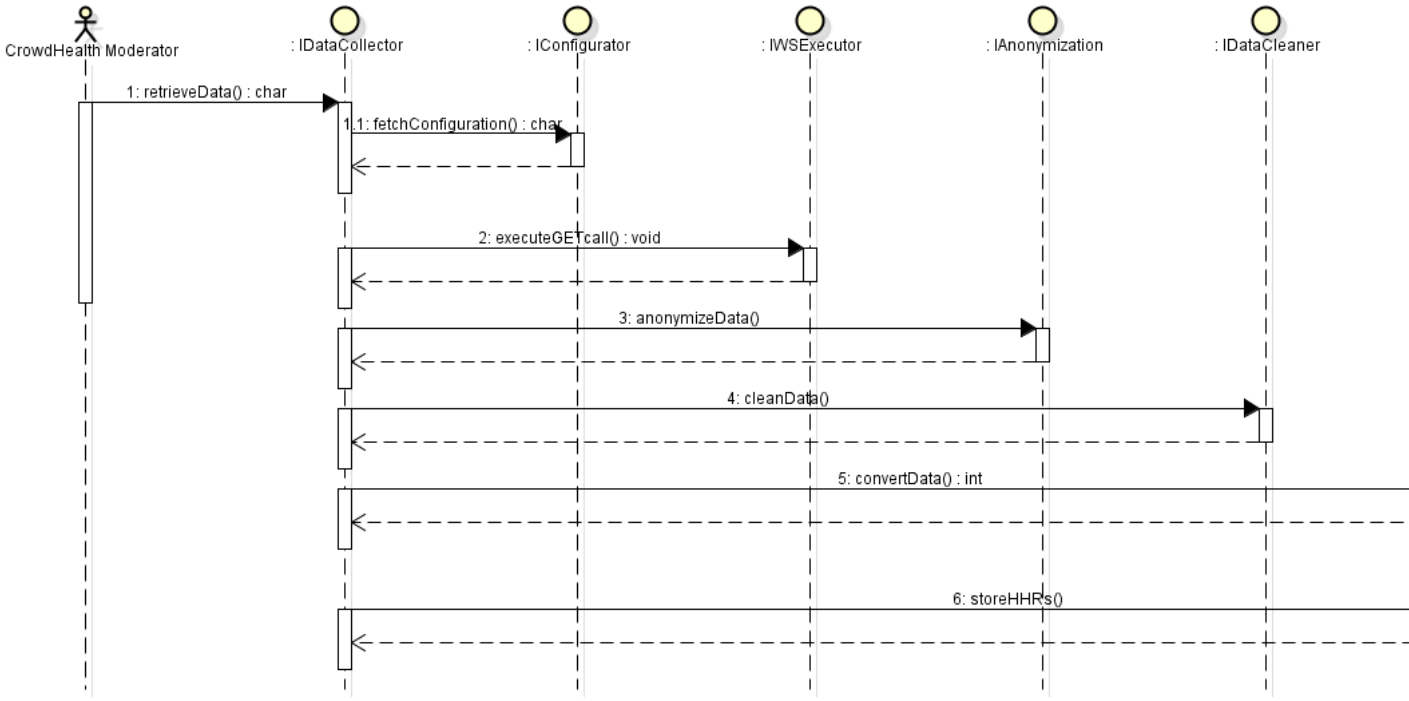

Figure 2-3), which in turn proceeds with the actual cleaning of the collected data and returns the cleaned data to the CrowdHEALTH Data Sources and Gateways Service.

Following the anonymization and the cleaning, the CrowdHEALTH Data Sources and Gateways Service triggers the interface exposed by the Data Converter component implementing the CrowdHEALTH data conversion approach, according to the FHIR standard (IDataConverter as per Figure 2-2 and

This project has received funding from the European Union's Horizon 2020 Programme (H2020-SC1-2016-CNECT) under Grant Agreement No. 727560

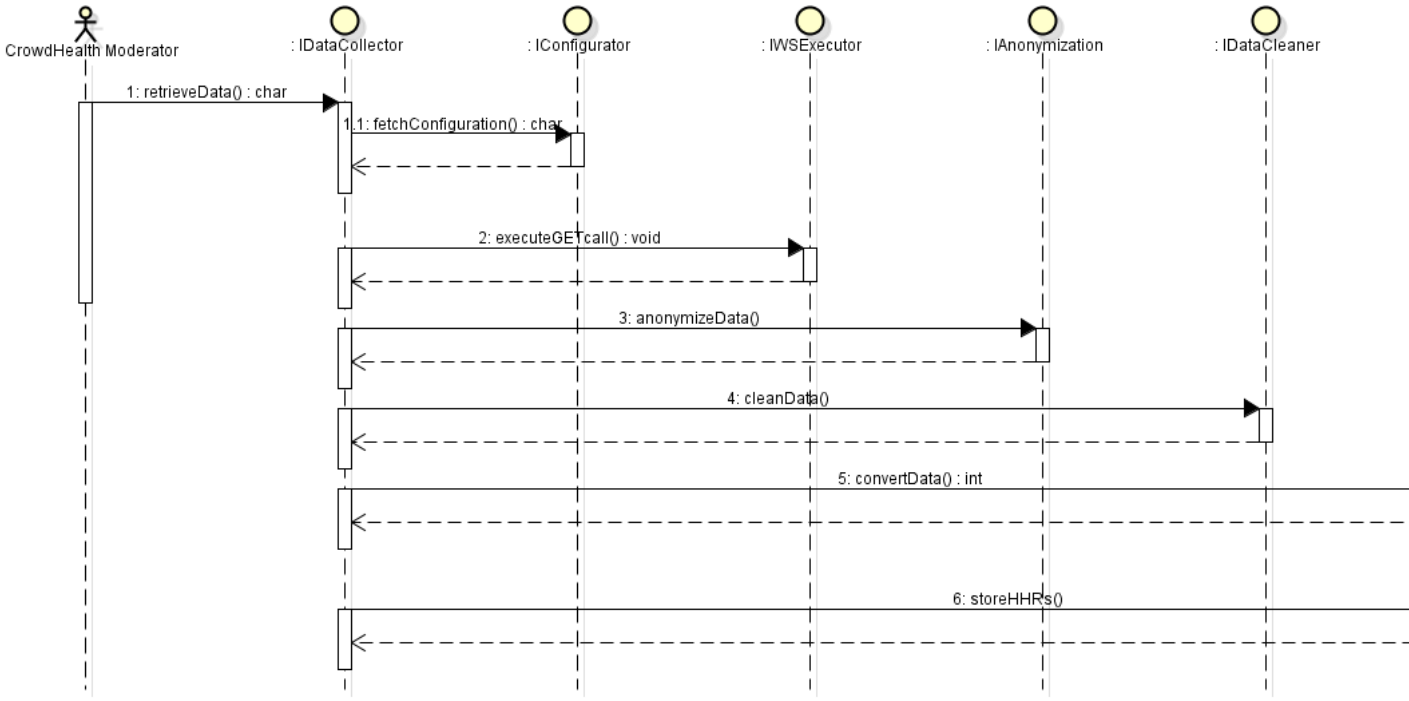

Figure 2-3), which in turn proceeds with the actual conversion of the collected, anonymised and cleaned data to FHIR compliant data, and returns them to the CrowdHEALTH Data Sources and Gateways Service for storing.

The last step in the process is associated with the actual storage of the collected, anonymised, cleaned, and FHIR-based converted data to the CrowdHEALTH platform. Thus, the CrowdHEALTH Data Sources and Gateways Service triggers the interface exposed by the component responsible for storing the HHR records (IHHRStoring as per Figure 2-2 and

This project has received funding from the European Union's Horizon 2020 Programme (H2020-SC1-2016-CNECT) under Grant Agreement No. 727560

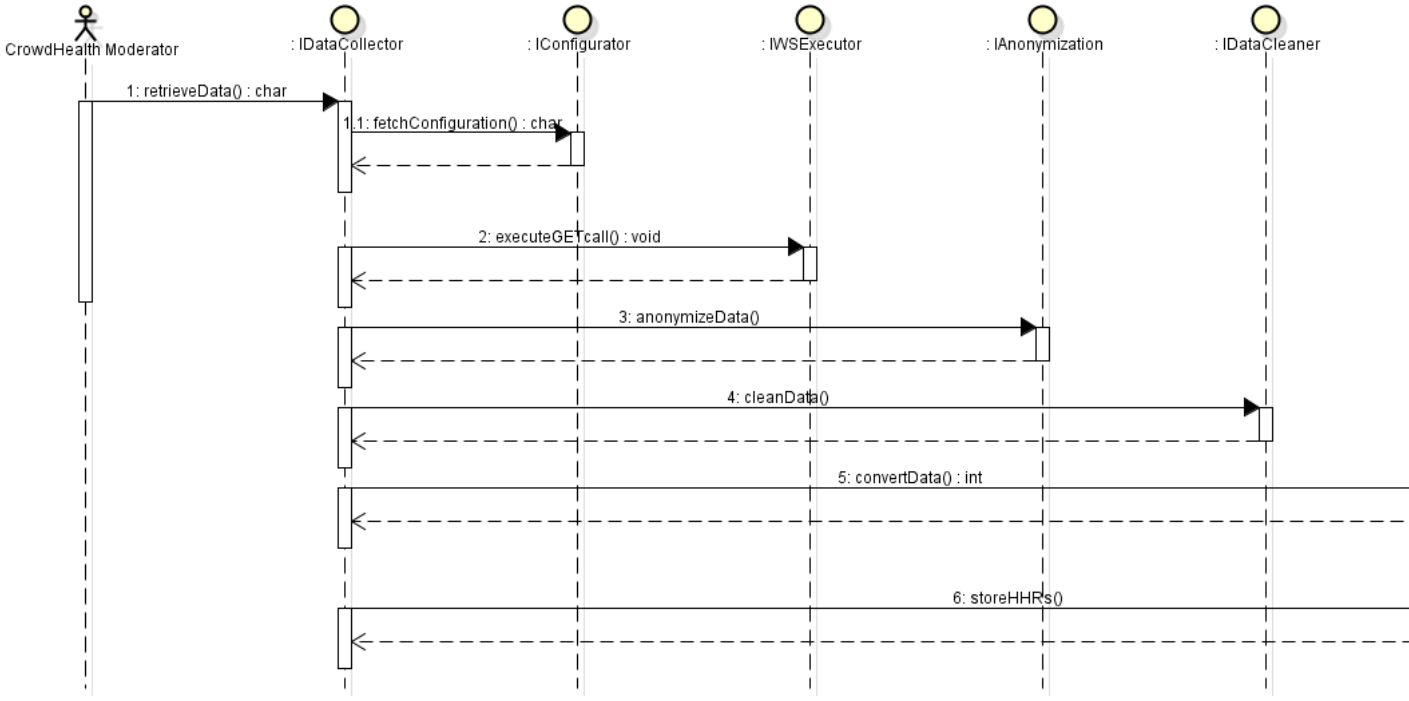

Figure 2-3), which in turn proceeds with the actual storage of the collected data.

Note: It is not within the context of the current deliverable to describe the interfaces exposed by the other platform components, since they will be described in the corresponding deliverables. Towards this end, the current deliverable is restricted to describing the roles of the internal components of the Data Sources and Gateways component and the internal interfaces facilitating the communication between the central, CrowdHEALTH Data Sources and Gateways Service, and the rest of the internal services implemented.

2.2.1. CrowdHEALTH Data Sources and Gateways Service

The CrowdHEALTH Data Sources and Gateways Service is responsible for handling all incoming and outgoing traffic on the CrowdHEALTH platform. It is responsible for mediating between the internal components (configuration service, DB connection handling service etc.) and the IDataCollector interface for information retrieval from the data providers. It is also responsible for handling scheduled information retrieval according to the specified communication frequency, as well as for orchestrating information processing as analysed in section 2.2, including communication with the other internal platform components (e.g. Data Cleaner, Data Converter etc. through the corresponding interfaces exposed by these components.

The CrowdHEALTH Data Sources and Gateways Service exposes the IDataCollector interface which supports two main functions:

- retrieveData: The retrieveData function is triggered internally and is executed in the case of pulling information from a data provider (e.g. connection to an API exposed by a data provider).
- receiveData: The receiveData function is triggered externally and is executed in the case of information being pushed from a data provider (e.g. an excel file pushed to the Data Sources and Gateways service to be parsed).

2.2.2. Configuration Service

The CrowdHEALTH Configuration Service is responsible for retrieving the data provider identifier from the CrowdHEALTH Data Sources and Gateways service, and for forwarding to the CrowdHEALTH Data Sources and Gateways Service the corresponding configuration file.

The CrowdHEALTH Configuration Service exposes the IConfigurator interface (see Table 3-1 in section 3.1) which supports two main functions:

- fetchConfiguration: The fetchConfiguration function is triggered internally and is executed in the case of pulling information from a data provider (e.g. connection to an API exposed by a data provider). The fetchConfiguration service expects as input the data provider identifier and returns the corresponding configuration file.
- SaveConfiguration. The saveConfiguration function is triggered internally and is executed when a new configuration for an existing or for a new data provider needs to be saved.

2.2.3. DB Connection Handling Service

The DB Connection Handling Service is responsible for undertaking the retrieval of information from databases, once the connection parameters (e.g. host, port, credentials, query, etc.) have been specified and communicated to the CrowdHEALTH Data Sources and Gateways Service.

The DB Connection Handling Service exposes the IDBConnectionHandler interface which supports (currently) three main functions, but could easily be extended to support additional functions, according to the types of Databases that may need to be supported in the context of the project:

- connectToMySQLDB: The connectToMySQLDB function is triggered internally and is executed in the case of pulling information from a data provider (i.e. connection to MySQL). The connectToMySQLDB expects as input the information included in the corresponding configuration file (see Table 3-2 in section 3.2).
- connectToOracleDB: The connectToOracleDB function is triggered internally and is executed in the case of pulling information from a data provider (i.e. connection to Oracle). The connectToOracleDB expects as input the information included in the corresponding configuration file (see Table 3-2 in section 3.2).

- connectToMongoDB: The connectToMongoDB function is triggered internally and is executed in the case of pulling information from a data provider (e.g. connection to Mongo Database). The connectToMongoDB expects as input the information included in the corresponding configuration file (see Table 3-2 in section 3.2).

2.2.4. File Parser Service

The File Parsing Service is responsible for undertaking the retrieval of information from files (e.g. csv files), once the connection parameters (e.g. file type, delimiter, file path, etc.) have been specified and communicated to the CrowdHEALTH Data Sources and Gateways Service.

The CrowdHEALTH File Parsing Service exposes the IFileParser interface which supports one main function:

- parseFile: The parseFile function is triggered internally and is executed in the case of information being pulled from or pushed by a data provider. The parseFile service expects as input the information included in the corresponding configuration file so as to know how to parse the file (see Table 3-3 in section 3.3).

2.2.5. RESTful Client Service

The RESTful Client Service is responsible for undertaking the retrieval of information from external exposed APIs, once the connection parameters (e.g. web service type, http method, URI, etc.) have been specified and communicated to the CrowdHEALTH Data Sources and Gateways Service.

The RESTful Client Service exposes the IWSExecutor interface which supports three main functions:

- executePOSTCall: The executePOSTCall function is triggered internally and is executed in the case of POSTing information from a data provider. The executePOSTCall expects as input the information included in the corresponding configuration file (see Table 3-4 in section 3.4).
- executePUTCall: The executePUTCall function is triggered internally and is executed in the case of PUTing information from a data provider. The executePUTCall expects as input the information included in the corresponding configuration file (see Table 3-4 in section 3.4).
- executeGETCall: The executeGETCall function is triggered internally and is executed in the case of GETting information from a data provider. The executeGETCall expects as input the information included in the corresponding configuration file (see Table 3-4 in section 3.4).

2.3. Data Sources and Gateways Component Implementation

From the data collection perspective, the Data Sources and Gateways component comprises (amongst others) a data acquisition framework. On top of the data acquisition, it also requires to orchestrate activities with the other internal components of the CrowdHEALTH platform (e.g. to send the data for cleaning and for conversion). With regards to the data acquisition in specific, the CrowdHEALTH consortium does not need to reinvent the wheel, since there is a plethora of data acquisition and data integration frameworks available, both open source and commercial, with Kettle probably being the most prominent one currently (Pentaho, 2017). Towards this end, the CrowdHEALTH consortium partners will explore the option of stripping the Kettle framework to support the functionalities aspired for the data acquisition and parsing, and customise it and extend it accordingly in order to support all additional requirements expected by the CrowdHEALTH platform, as also documented in section 2.1.

3. Data Sources and Gateways Component Attributes

The current chapter provides the attributes for the four different services aforementioned, supported by the Data Sources & Gateways component. The attributes have been intentionally left blank for two main reasons:

1. The data sources per se have not been made available by the time of writing of the deliverable, thus only the templates used for the collection of the necessary information have been provided.
2. Even after the information has been made available by the project pilot partners, given the privacy of the information provided and the public nature of the current deliverable, this information will not be disclosed.

3.1. Configuration Service

The current subchapter provides the attributes for the configuration service. The information to be provided includes the following fields as identified in Table 3-1.

Required Field	Field Attribute
Identifier (Data Provider)	e.g. BIO, CRA, KI, etc.
Dataset Identifier (Optional)	e.g. Biosignals, Allergies, etc.

Table 3-1 Configuration Service Attributes Specification

3.2. DB Connection Handling Service

The current subchapter provides the attributes for the DB connection handling service. The information to be provided includes the following fields as identified in Table 3-2.

Required Field	Field Attribute
DB Type	e.g. MySQL, Oracle, Mongo, etc.
Host	e.g. 192.168.X.X
Port	e.g. 3308
DB Name	e.g. Biosignals
Username	e.g. Kostas
Password	e.g. eHTb7%Pxa9
Query	e.g. SELECT * FROM Allergies WHERE n = 'lactose' ORDER BY Age;

Table 3-2. DB Connection Handling Service Attributes

3.3. File Parsing Service

The current subchapter provides the attributes for the file parsing service. The information to be provided includes the following fields as identified in Table 3-3.

Required Field	Field Attribute
Type of File	e.g. csv, etc.
Delimiter	e.g. , or ;
Line Header	e.g. id;patient_id;
File Path (in case of pulling file)	

Table 3-3. File Parsing Service Attributes

3.4. RESTful Client Service

The current subchapter provides the attributes for the RESTful client service. The information to be provided includes the following fields as identified in Table 3-4.

Required Field	Field Attribute
Web Service Type	e.g. REST
File Type	e.g. XML, JSON, etc.
HTTP Method	e.g. POST
Authentication	e.g. Authentication=gjdf84ngs87gfsd (HTTP Header)
URI	e.g. /api/patient/567868
Request Body (if available)	e.g. {“patient_id”:567868}
Response Codes	e.g. 200 (OK), 201 (Created), 401 (Unauthorised) etc.

Table 3-4. RESTful Client Service Attributes

4. Conclusions

The scope of D3.5 was to document the architecture and design of the Data Sources and Gateways component and the envisioned abstracted API for all information sources handled within the context of CrowdHEALTH. The initial version of the component design has built upon the requirements collected from the use case participants acting also as data providers.

The second chapter outlined the component design, providing information regarding the component scope, and the initial design of the component, illustrating its relation to the other components of the overall architecture. The CrowdHEALTH Data Sources and Gateways Service comprises of four main services: 1) The Configuration Service, 2) The DB Connection Handling Service, 3) The File Parsing Service and 4) The RESTful Client Service. These four services are not arbitrary. On the contrary, these services were designed based upon the specifications of the project pilots.

Chapter 3 undertook the analysis of the specification of the Data Sources and Gateways component, documenting the specification of the internal subcomponents of the Data Sources and Gateways component which will drive its implementation. Since the data sources per se have not been made available, the current version of the deliverable documented the templates used for the collection of the necessary information.

The design and specifications of the Data Sources and Gateways component are subject to changes that will be documented in the second version of the current deliverable. More specifically, the current version will guide the implementation of the first version of the Data Sources and Gateways component software prototype, which in turn will be evaluated within the context of the project and enriched in order to meet additional end user requirements or just be refined in its second and final version.

ANNEX A. Connection Specifications from Pilot Partners

The specifications were collected using the template indicated in Table 0-1 and Table 0-2. The aforementioned filled-in tables constitute the consolidation of the information provided by the pilot partners.

As per the template, the information requested from the project partners includes the following information:

- **Type of Data Source**: Represents the type of the source through which information needs to be retrieved. Possible values include SQL DB (e.g. MySQL), NoSQL DB (e.g. MongoDB), exposed API, etc.
- **Connection to Data Source**: Represents the type of connection to the data source. Possible values include implementation of API, DB access or provision of file path.
- **Access to Data Source**: Represents whether access to the information provided is publicly accessible, or private, thus requiring some kind of authentication.
- **Communication Type**: Represents the style of communication. Possible values are either a) push, where the request for a given transaction is initiated by the publisher or central server, or b) pull, where the request for the transmission of information is initiated by the receiver or client.
- **Communication Frequency**: Represents the frequency of information retrieval, and varies per use case partner but also among the different datasets from the same data provider.
- **Authentication**: Represents the security in communication. Possible values include Username / Password, Token, etc.
- **Compliance to the FHIR standard:** Represents compliance to the FHIR standard (HL7 International, n.d.). Possible values include Yes (if the data comply with the FHIR standard) or No if the data do not comply with it (regardless whether they comply with another standard or are stored in a custom format.
- **Record Structure**: Represents a data source record example with column names (in case of DB) or field names (in case of APIs).
- **Unique Identifier**: Represents a unique identifier (Unique ID column name (in case of DB) or field name (in case of APIs)) in order to distinguish subjects within the same dataset.
- **Size**: Represents the volume of information currently available and aspired to be integrated in the CrowdHEALTH platform.

#	Question	BIO Biosignals	BIO PHR	BIO Allergies	BIO Medication	BIO Social
1	Type of Data Source	API	API	API	API	SQL DB
2	Connection to Data Source	Implement API	Implement API	Implement API	Implement API	Implement API
3	Access to Data Source	Private	Private	Private	Private	Private
4	Communication Type	Pull	Pull	Pull	Pull	Pull
5	Communication Frequency	TBD	Daily or less often	Daily or less often	Daily or less often	TBD
6	Authentication	Token	Token	Token	Token	Token
7	Compliance to FHIR	Yes	Yes	Yes	Yes	No
8	Record Structure	FHIR Observation	FHIR Observation	FHIR AllergyIntolerance	FHIR Medication	Span across DB tables
9	Unique Identifier	uuid	uuid	uuid	uuid	N/A

Table 0-1. Pilot Gateways Part A

#	Question	CRA	DFKI Various	HULAFE	KI	ULJ
1	Type of Data Source	SQL DB	API	SQL DB	SQL DB	Excel
2	Connection to Data Source	Implement API	Implement API	Implement API	Allow DB access	Allow DB access
3	Access to Data Source	Private	Private	Private	Private	Private
4	Communication Type	Push	Push	Push	Pull	Pull
5	Communication Frequency	Every 2-3 months	Real time	15 days	Weekly	Annual
6	Authentication	Username / Password	Other	Username / Password	Username / Password	Username / Password
7	Compliance to FHIR	Yes	No	No	No	No

#	Question	CRA	DFKI Various	HULAFE	KI	ULJ
8	Record Structure	Non-disclosable example provided	Non-disclosable example provided	Non-disclosable example provided	Non-disclosable example provided	Non-disclosable example provided
9	Unique Identifier	To be confirmed	Depending on data source. See DFKI.xlsx for details	PatientID	CRID	CROWD_ID

Table 0-2. Pilot Gateways Part B

References

1. HL7 International. (n.d.). FHIR Standard. Retrieved October 24, 2017, from https://www.hl7.org/fhir/

2. Kyriazis, D., Maglogiannis, I., Xenakis, C., Mavrogiorgou, A., Kiourtis, A., Peppas, G., … Stanimirovic, D. (2017). D2.1 State of the art and requirements analysis v1. *EC H2020 CrowdHEALTH Project*.

3. Pentaho. (2017). Kettle. Retrieved October 24, 2017, from https://github.com/pentaho/pentaho-kettle