ORIGINAL PAPER

Data Sources and Gateways: Design and Open Specification

Konstantinos Perakis[1],
Dimitris Miltiadou[1], Antonio
De Nigro[2], Francesco Torelli[2],
Lydia Montandon[3], Andriana
Magdalinou[4], Argyro
Mavrogiorgou[5], Dimosthenis
Kyriazis[5]

[1]Singular Logic, Cyprus

[2]Engineering-Ingegneria Informatica,

[3]Atos, Madrid, Spain

[4]European Federation for Medical
Informatics, Lausanne, Switzerland

[5]University of Piraeus, Piraeus, Greece

Corresponding author: Konstantinos Perakis,
Singular Logic, Cyprus. E-mail: kperakis@
ep.singularlogic.eu. ORCID ID:https://orcid.
org/0000-0002-2236-9816

ABSTRACT

Introduction: With With the proliferation of available ICT services, several sensors and health applications have become ubiquitous, while many applications have been developed to detect certain health conditions and early signs of disease. Currently, all these services operate independently, and the available data is heterogeneous with limited value gained from its exploitation. **Aim:** The Data Sources and Gateways component aims at providing an abstracted and unified API to support the data accumulation from various sources including healthcare organisations, biosensors, laboratories and mobile applications. Meanwhile it tackles connectivity and communication issues with such information sources. **Methods:** The CrowdHEALTH Data Sources and Gateways Service incorporates four main services: The Configuration Service, The DB Connection Handling Service, The File Parsing Service and The RESTful Client Service. **Results:** The initial version of the component design has built upon the requirements collected from the use case participants acting also as data providers. **Conclusion:** These four services presented in this paper guide the implementation of the first version of the Data Sources and Gateways component software prototype. The Data Sources and Gateways component remains to be evaluated within the context of the project and be enriched in order to meet additional end user needs.

Keywords: Data sources, Data acquisition.

1. INTRODUCTION

With the proliferation of available ICT services, several sensors and health applications have become ubiquitous, while many applications (1) have been developed to detect certain health conditions and early signs of disease (2). Currently, all these services (3) operate independently, and the available data is heterogeneous with limited value gained from its exploitation (4). Applications range from smartphone-based wellness applications to fitness bands and smartwatches, all accumulating real time health related data. However, these implementations are device specific and there are limitations in the incorporation of diverse data sources. Therefore, data from isolated data sources cannot give the big picture of a user's health status or provide a holistic view of a user's lifestyle (5). Within this context, health care requires intensive use of information technology to collect and analyze data and then manage and exploit the knowledge derived from the data (6). With regards to the data acquisition in specific, 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 (7). The scope of the Data Sources and Gateways component is to deliver a software tool for supporting the process of acquiring multimodal data from various sources and providers, which may be described in different and often unstructured or inconsistent formats. The Data Sources and Gateways component aims at providing an abstracted and unified API which supports the acquisition of information from sources including healthcare organisations, biosensors, laboratories and mobile applications. Meanwhile it aims at resolving connectivity and communication issues related to the information sources, ensuring, through interaction with the rest of the internal components of the CrowdHEALTH architecture, the integration of the syntactically harmonized, cleaned and validated information into the Holistic Health Records (8).

2. AIM

The aim of this article is to outline the architecture and initial design of the data sources and gateways framework. This framework is developed to enable the acquisition of multimodal data from different sources and various data source providers and solve current connectivity and communication issues.

3. METHODS

The The CrowdHEALTH Data Sources and Gateways Service comprises of four main services: The Configuration Service, The DB Connection Handling Service, The File Parsing Service and The RESTful Client Service. These four services were designed based upon the specifications of the project pilots. The specifications were collected using a template. For each template, the information requested from the project partners includes the following information:

· Type of Data Source: Represents the type of 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 ac-

cessible, 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 of 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..

4. RESULTS

The Data Sources and Gateways component have the following requirements as identified in D2.1 State of the art and requirements analysis v1. (9):

· Facilitate the connection to an appropriately specified (SQL or No-SQL) Database, for the retrieval of the information.

· Facilitate the connection to an appropriately specified API, for the retrieval of the information.

· Facilitate the parsing of files (e.g. excel or csv files, for the retrieval of the information).

· Provide access to a configuration service, facilitating the configuration of the connection parameters per connection type and source.

· Support pulling data from external data sources (e.g. through REST APIs) per predefined time intervals.

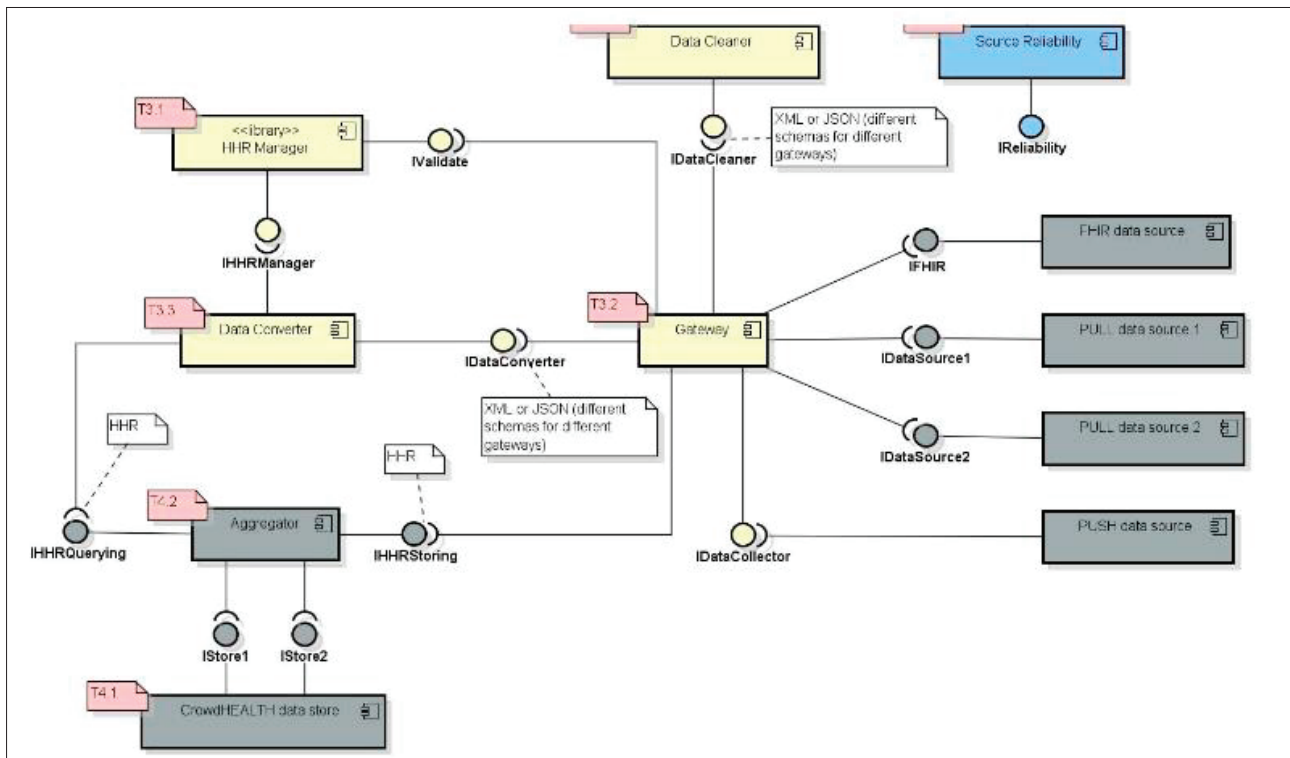· Support data from external data sources being

Figure 1.Graphical representation of the positioning of the CrowdHEALTH Data Sources and Gateways component
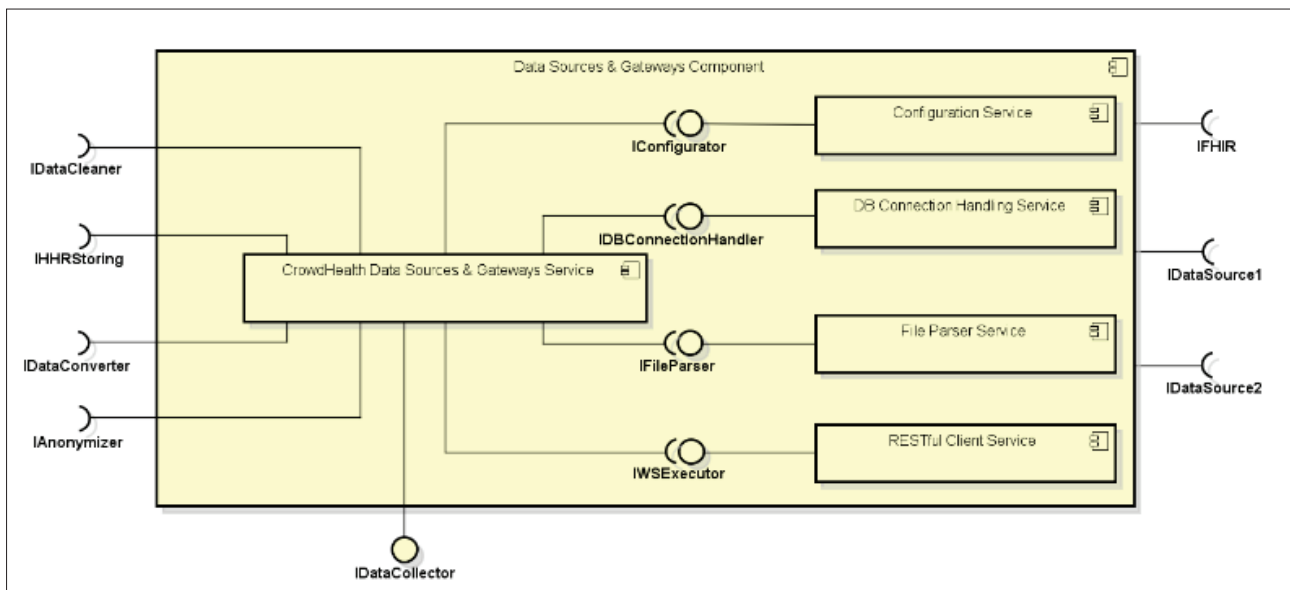

Figure 2. Graphical representation of the Data Sources and Gatewayscomponent

pushed to the platform per predefined intervals.

· Support token-based authentication with the data sources to safeguard data integrity and non-repudiation.

· Support username and password-based authentication with the data sources to safeguard data integrity and non-repudiation.

· Facilitate a standardised connection to other internal components of the CrowdHEALTH platform, such as the Data Cleaner, the Data Converter, etc.

· Facilitate the connection to unknown, plug 'n play sources, mapping them to already known sources in order to identify the information types made available.

The Figure 1 provides a graphical representation of the positioning of the CrowdHEALTH Data Sources and Gateways component within the holistic project approach. As presented in 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,
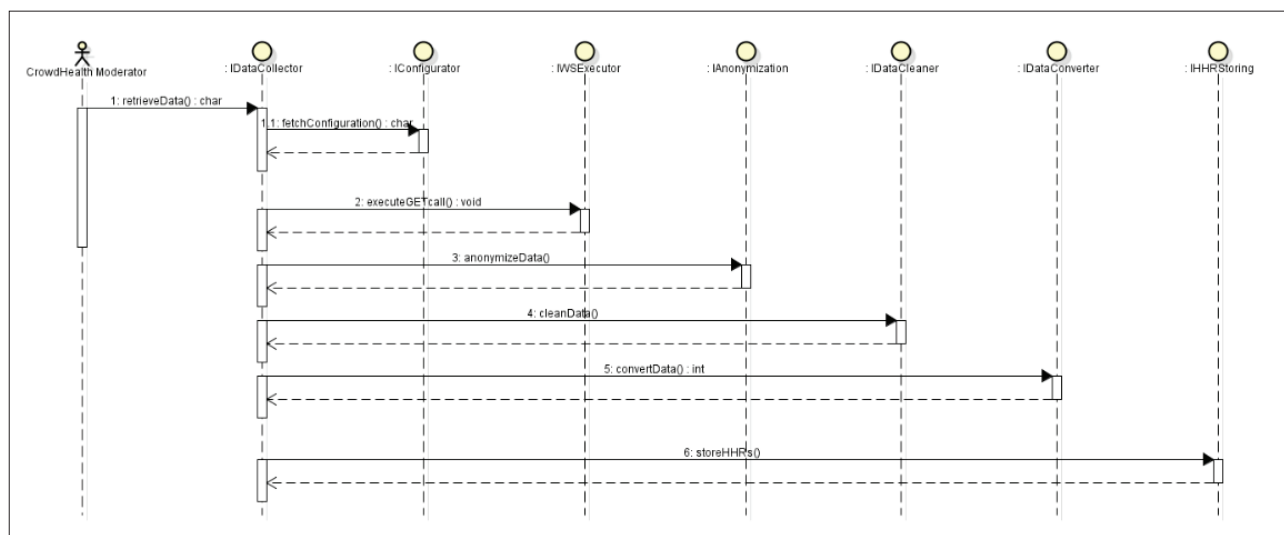
Figure 3. Information flow and the procedures and processes supported by the CrowdHEALTHData Sources and Gateways component

and IDataConverter) so that aggregated information can be forwarded to these components.

Figure 2 provides a graphical representation of the Data Sources and Gateways component, highlighting its internal subcomponents, as well as its interfaces with other internal components of the CrowdHEALTH platform. The graphical representation in Figure 2 incorporates 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. 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. 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 GatewaysService, which can be perceived as the orchestrator of the various services. The central stripe (including the CrowdHEALTH Data Sources & Gateways Service and the IDataCollector interface) represents the central CrowdHEALTH Data Sources and GatewaysService which is considered to be responsible for managing all incoming and outgoing traffic on the CrowdHEALTH platform, while providing connection options. The Crowd-HEALTH Data Sources and GatewaysService is the mediator between the internal components (configuration service, DB Connection Handling Ser-

vice etc.) and for exposing the IDataCollector interface so as to facilitate information retrieval from the data providers. The external left stripe (including the IDataCleaner, theIHHRStoring, the IDataConverter and the IAnonymizer interface) represents the interfaces that are external to the CrowdHEALTH Data Sources and Gateways Components. The CrowdHEALTH Data Sources and GatewaysService comprises of four main services as depicted in Figure 2. The Configuration Service, The DB Connection Handling Service, The File Parsing Service, The RESTful Client Service. These four services were designed based upon the specifications of the project's pilots. The envisioned information flow and the procedures and processes supported by the CrowdHEALTH Data Sources and Gateways component are illustrated in Figure 3. As can be seen in Figure 3, the Data Collector interface (IDataCollector), which is the only interface exposed to the rest of the Crowd-HEALTH 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 forwards the request to the Configuration Service, and the Configuration Service responds with the appropriate configuration file. Upon receipt of the connection configuration file, the CrowdHEALTH Data Sources and GatewaysService through the IDataCollector interface initiates information re-

trieval, connecting, for example, to the external API provided by the data providers, invoking the corresponding internal RESTful Client Service through the internal IWSExecutor. The information needs to be further anonymized after the retrieval, 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 GatewaysService triggers the interface exposed by the Anonymization component implementing the CrowdHEALTH anonymization approach (IAnonymizer in Figure 2 and Figure 3), which proceeds with the actual anonymization of the collected data and returns the anonymized data to the CrowdHEALTH Data Sources and GatewaysService. Upon anonymization, the CrowdHEALTHData Sources and GatewaysService triggers the interface exposed by the Data Cleansing component implementing the CrowdHEALTH data cleaning approach (IDataCleaner in Figure 2 and Figure 3), which in turn proceeds with the actual cleaning of the collected data and returns the cleaned data to the CrowdHEALTH Data Sources and GatewaysService. After this stage, the CrowdHEALTH Data Sources and GatewaysService triggers the interface exposed by the Data Converter component implementing the CrowdHEALTH data conversion approach, according to the FHIR standard (IDataConverter as in Figure 2 and Figure 3), which proceeds with the actual conversion of the collected, anonymized and cleaned data to FHIR compliant data, and returns them to the CrowdHEALTH Data Sources and GatewaysService for storing. The last phase in the process is associated with the actual storage of the collected, anonymized, cleaned, and FHIR-based converted data to the CrowdHEALTH platform.

4.1. CrowdHEALTH Data Sources and GatewaysService

The CrowdHEALTH Data Sources and GatewaysService is responsible for handling 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 collecting information 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, 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 GatewaysService exposes the IDataCollector interface which supports two main functions:

· retrieveData: The retrieve Data function is triggered internally and is executed in the case of pulling information from a data source (e.g. connection to an API exposed by a data provider).

· receiveData: The receive Data function is triggered externally and is executed in the case of information being pushed by a data provider (e.g. an excel file pushed to the Data Sources and GatewaysService to be parsed)

4.2. Configuration Service

The CrowdHEALTH Configuration Service is responsible for retrieving the data provider identifier from the CrowdHEALTH Data Sources and Gatewaysservice, and for forwarding to the CrowdHEALTH Data Sources and GatewaysService the corresponding configuration file. The CrowdHEALTH Configuration Service exposes the IConfigurator interface which supports two main functions:

· fetchConfiguration: The fetchConfiguration function is triggered internally and is executed in the case of pulling information by 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.

4.3. DB Connection Handling Service

The DB Connection Handling Service undertakes 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 GatewaysService. The DB Connection Handling Service exposes the IDBConnection Handler interface which supports up until now three main functions, but could be extended to support additional operations, 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 connectTo-

MySQLDB expects as input the information included in the corresponding configuration.

· fileconnectToOracleDB: 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.

· 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.

4.4. File Parser Service

The File ParsingService undertakes the information retrieval 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 GatewaysService. The CrowdHEALTH File ParsingService exposes the IFileParser interface which supports one main function:

· parseFile: The parseFile 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.

· 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.

· 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.

5. CONCLUSION

The presented version of the Data Sources and Gateways component design has been developed after taking into consideration the requirements gathered from the use case participants. The CrowdHEALTH Data Sources and Gateways Service consist of four 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, but they were designed based upon the specifications of the project's pilots. More specifically, the current model guides the implementation of the first version of the Data Sources and Gateways component software prototype. The prototype has to be evaluated within the context of the project and include additional end user requirements..

REFERENCES

1. Mena LJ, Felix VG, Ostos R, Gonzalez JA, Cervantes A, Ochoa A, et al. Mobile personal health system for ambulatory blood pressure monitoring. Computational and mathematical methods in medicine. 2013; 13. doi: 10.1155/2013/598196.

2. Mantas J. Future trends in Health Informatics - theoretical and practical. Studies in health technology and informatics. 2004; 109: 114-127.

3. Liaskos J, Frigas A, Antypas K, Zikos D, Diomidous M, Mantas J. Promoting interprofessional education in health sector within the European Interprofessional Education Network. International Journal of Medical Informatics. 2009; 78: S43-S47.

4. Kiourtis A, Mavrogiorgou A, Menychtas A, Maglogiannis I, Kyriazis D. Structurally Mapping Healthcare Data to HL7 FHIR through Ontology Alignment. Journal of Medical Systems. 2019; 43: 62. doi:https://doi.org/10.1007/s10916-019-1183-y

5. Amin M, Banos O, Khan W, Bilal H, Gong J, Bui DM. et al. On Curating Multimodal Sensory Data for Health and Wellness Platforms. Sensors. 2016; 16(7): 980. doi: https://doi.org/10.3390/s16070980

6. De Georgia M, Kaffashi F, Jacono F, Loparo K. Information Technology in Critical Care: Review of Monitoring and Data Acquisition Systems for Patient Care and Research. The Scientific World Journal. 2015; 9.

7. Pentaho. Kettle. 2017 [online]https://github.com/pentaho/pentaho-kettle

8. Montandon L, Kyriazis D, Ramon Valero Z, Fernandez Llatas C, Traver V. CrowdHEALTH- Collective wisdom driving public health policies. 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS). 2019; 1-3..

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