D4.7

Design of Health Record Index

ABSTRACT

Electronic Health Record (EHR) data is being demanded to be shaped into a common standardized and interoperable format, and be both easily stored and securely accessed from anywhere. Currently, this is achieved through the secure storage of the EHR data in different cloud repositories, which however increase the difficulty of accessing it in emergency situations from healthcare practitioners, or even from the citizens' themselves. The latter need to have specific credentials for accessing EHR data in private cloud repositories, which can be almost impossible in urgent situations where this data must be accessed no matter what. For that reason, in this deliverable a health record indexing methodology is being proposed that facilitates the access of non-privileged certified users, to the EHR data stored in cloud repositories of citizens-in-need, under emergency cases. In this context, the research that has been conducted prior to specifying and designing the proposed HR index is being provided, including a detailed study of data and health data indexing techniques and methodologies. Finally, the overall vision, scope and usage of the HR index is provided in the context of an Emergency scenario, concluding to, the design of the HR index, including its components and interfaces.

Delivery Date	22nd January 2021
Work Package	WP4
Task	T4.2
Dissemination Level	Public
Type of Deliverable	Report
Lead partner	UPRC

InteropEHRate project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106.

This document has been produced in the context of the InteropEHRate Project which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106. All information provided in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose.

CONTRIBUTORS

	Name	Partner
Contributors	Thanos Kiourtis, Argyro Mavrogiorgou, Aikaterini Aggeli, Charis Tsigkounis	UPRC
Contributors	Chrysostomos Symvoulidis, Stella Dimopoulou, Konstantinos Koutsoukos	BYTE
Reviewers	Stephan Schug	EHTEL
Reviewers	Patrick Duflot	CHU

LOG TABLE

Version	Date	Change	Author	Partner
0.1	2020-09-03	ToC circulated	Thanos Kiourtis, Argyro Mavrogiorgou	UPRC
0.2	2020-09-07	Finalized Section 1, Provided input regarding the scope and the design of the HR index in Section 3, finalized Section 4	Thanos Kiourtis, Aikaterini Aggeli, Argyro Mavrogiorgou	UPRC
0.3	2020-09-15	Finalized input from partners and Proofreading performed	Thanos Kiourtis, Argyro Mavrogiorgou, Charis Tsigkounis, Chrysostomos Symvoulidis, Stella Dimopoulou	UPRC, BYTE
0.4	2020-09-18	Fixed images, tables & references	Thanos Kiourtis, Argyro Mavrogiorgou	UPRC
0.5	2020-09-21	Finalized and sent for internal review	Thanos Kiourtis, Argyro Mavrogiorgou	UPRC
0.6	2020-12-13	Addressed reviewers' comments, including updates in the design of the libraries	Thanos Kiourtis, Argyro Mavrogiorgou, Chrysostomos	UPRC, BYTE

			Symvoulidis	
0.7	2021-01-04	Sent for Quality Check	Thanos Kiourtis	UPRC
0.8	2021-01-04	Quality Check	Argyro Mavrogiorgou	UPRC
0.9	2021-01-21	Updates on Section 3 regarding the sequence diagrams and the remote interfaces	Thanos Kiourtis	UPRC
Vfinal	2021-01-22	Final check and submission	Francesco Torelli Laura Pucci	ENG

ACRONYMS

Acronym	Term and definition
EHR	Electronic Health Record
HR index	Health Record index
S-EHR cloud	Smart Electronic Health Record cloud
HCP	Health Care Practitioner
S-EHR	Smart Electronic Health Record
IR	Information Retrieval
F-MTI	French Multi-Terminology Indexer
CB-PHR	Cloud-based Personal Health Record systems
PHR	Personal Health Record
QR code	Quick Read code

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Scope of the document

In the field of electronic health and mostly in emergency situations, the exchange of data between citizens - patients and healthcare professionals, is of major importance. Based on specific research, it was observed that what is mostly missing in the aforementioned domain, is the option of the citizens to store their Electronic Health Record (EHR) data on cloud repositories, and having the ability to provide to their healthcare practitioners the means of accessing these data without specific credentials and without providing them a direct access (i.e. link) to their cloud repository, for security and privacy reasons.

The purpose of this deliverable is to propose an innovative intermediate in the form of a Health Record index (HR index), for informing the healthcare practitioners about the cloud location of the stored EHR data without directly providing the data to them. The latter will facilitate emergency cases where the citizens are not able to provide their consent for third-parties that need to access their EHR data (supposing that in the past such a consent has been already approved by the citizens for granting access to the stored data to healthcare practitioners with pre-specified data access tokens).

Consequently, the main focus of this document is to define the technical specifications of the proposed HR index, providing also detailed descriptions of its context of use, outlining a detailed description of its functionality, accompanied by an explanation of its purpose of existence. Moreover, this deliverable describes the design of two libraries for the HR index. The first library is a Java-based component that can be nested in any Android application (Android version 4.3 or higher). It offers a set of Java operations for uploading data to the cloud storage (i.e. S-EHR Cloud), using the HR index. The second library is a Java-based component that can be embedded in any Java application (web or desktop applications). It offers a set of Java operations enabling the application (used by a HCP) to download the data of a citizen from the cloud (i.e. S-EHR Cloud), using the HR index functionalities.

1.2.Intended audience

The current document is mainly intended for developers, and application owners that are interested in designing and building either S-EHR applications or HCP applications, and who aim at exploiting and reusing the HR index and indexing techniques, in the context of their applications. As a result, this audience will be able to deliver the aforementioned functionality in their developed applications, since the HR index can be efficiently adopted by other systems and applications - as it will be available in the form of libraries. Apart from that, the document is intended for researchers as well, as they may be interested in understanding the way that the HR index works, influenced by it, and possibly extend and update it.

1.3.Structure of the document

The deliverable is organized in the following Sections:

- Section 1 (the current section) introduces the overall concept of the document, defining its scope, intended audience, and relation to the other project tasks and reports.
- Section 2 outlines the related research and work regarding health records indexing. An end-to-end example is provided, for the better understanding of the aforementioned methodology.

- Section 3 describes in full detail the purpose of the existence of the HR index, whereas stating the design of the HR index, in the form of external operations and the way that they are invoked.
- Section 4 outlines the conclusions of the current document, including future developments and updates of the design of the HR index.

1.4.Updates with respect to previous version (if any)

This is ”Not Applicable” since it is the first version of the deliverable.

2. RELATED WORK & DEFINITIONS

The purpose of Section 2 is to provide the research that has been conducted prior to specifying and designing the proposed HR index. A detailed study of domain-specific indexing methodologies, in the context of EHR data is being provided. The difficulties and weak points of these techniques are specified, concluding in an end-to-end example for making the purposes of the EHR data indexing clearly understandable.

2.1. Health Records Indexing

The purpose of Section 2.1 is to explain what is a Health Record indexing and why it should be used, providing information on health record indexing techniques, and finally concluding to an end-to-end health record indexing example, in order for the purpose of the indexing process to be more understandable.

2.1.1. Definition

Patient records contain crucial documents and data for managing the treatments and healthcare of patients in the hospital. Care providers waste precious time searching the patient records to collect all the important and useful information. Health record indexing is a very important function since it involves organizing and storing information such as the patient's demographic and treatment information together in one place for easy retrieval later. Cloud-based Personal Health Record systems have great potential in facilitating the management of individual health records. This helps medical staff to access all the information that they need quickly and easily, in order to determine treatment options as well as to be compliant with patient data storage and healthcare document management requirements.

2.1.2. Health Record Indexing Techniques & Research Performed

In the field of Health Record Indexing, there does not exist any similar work that performs the same actions and has the same usability as the Health Record Indexing technique that is proposed in the current deliverable. However, several similar researches and works have been performed in the healthcare domain, and described below, taking into consideration that many times specific information is scattered in numerous documents.

To begin with, in the work of [EHRLER 2007] the authors wanted to index the patient records and allow fast retrieval so they presented a methodological approach for using an Information Retrieval (IR) tool. They worked with many small corpora that were indexed independently. In a large collection of documents this tool can select a manageable number of documents to satisfy an expressed need for information. Before queries the system must create a representation of each document in the collection to accomplish the retrieval task. This representation consists of a table containing links that allow the knowledge of the terms that appear in the documents and with what frequency. During the search process, the system calculates the degree of match between the terms included in the index and a corresponding set of terms derived from the query. This degree of match is based on the frequency of the words in the document itself and in the corpus, providing the basis for deciding whether or not a document should be retrieved. In the same context, the authors in [YUAN 2016] proposed a cloud-based framework for privacy aware healthcare monitoring systems, allowing fast data retrieval and indexing with strong privacy assurance. For easy data sampling they exploited recent efforts on encrypted search and adopted compressive sensing. To design a novel encrypted index with high-performance customization to face the challenges of continuously generated medical data samples at high rates and large volumes, they also adopted efficient content-based indexing techniques and fine-grained locking algorithms. Through their approach, the authors improved building speed with non-trivial multi-thread support, achieved provable security, memory efficiency, explored the relationship between accuracy and efficiency and reduced the bandwidth of secure retrieval. Furthermore, in [PEREIRA 2008] the authors compared the lemmatization and stemming as methods to process French medical text for indexing. They developed the French Multi-Terminology Indexer (F-MTI), a MeSH automatic indexing tool, with the multi-terminology and stemming algorithm to assist the development of a French online health gateway. The indexing strategies were evaluated on a total of 18,814 resources that were indexed manually. The result was that there is a difference in the indexing performance when stemming or lemmatization is used. F-MTI was the first multi-terminology tool available for a language other than English. The language that was used plays a very important role in F-MTI's performance. Unfortunately, there were fewer UMLS semantic network mappings between MeSH and other terminologies in French because there were only 10 medical terminologies available in French while 100 are available in English. To this end, in [WAN 2019] the authors wanted to help process spatial queries efficiently so they proposed an energy and time-efficient multidimensional data indexing scheme which is designed to answer range queries. By range queries it can be retrieved stored data that satisfies a specific set of interval-based constraints. The authors evaluated its utility using simulations. Some data indexing methods that have been proposed to utilize hierarchical indexing structures, using binary space partitioning are k-means clustering, quad-tree, kd-tree, and Voronoi-based methods to provide more efficient routing with less latency. The results were that the Voronoi Diagram-based algorithm minimizes the average query response time and energy consumption. The Voronoi Diagram data index model is also suitable for general queries operations.

2.1.3. Example of Health Record Indexing

The example below depicts the usage of a Health Record Indexing mechanism, as described by the authors in [YAO 2018]. As the authors mention, in the generic Cloud-based Personal Health Record systems (CB-PHR) there are three kinds of entities: the cloud server, the data owners and the data providers. A data owner is a patient who owns the PHRs. A data provider can refer to any of his health providers such as a doctor or a hospital, a patient himself, even his personal health monitoring device. The cloud server stores and provides anywhere and anytime access to the PHRs submitted by the data providers of each data owner. The data providers must encrypt the PHRs before outsourcing them to the cloud server, because each data owner has strong privacy concerns for his PHRs. Each data provider uploads a data index to the cloud server to ensure efficient search for the encrypted PHRs. The data owner or any of his authorized data providers can submit data queries to the cloud server. Queries and data indexes should be encrypted to prevent information disclosure. The cloud server explores the data indexes to locate the PHRs that satisfy any query without the need to decrypt the data indexes, PHRs, or data queries. Finally, the cloud server returns the corresponding encrypted PHRs to the data user who can decrypt them with the correct decryption key. The aforementioned example is clearly visualized in Figure 1.

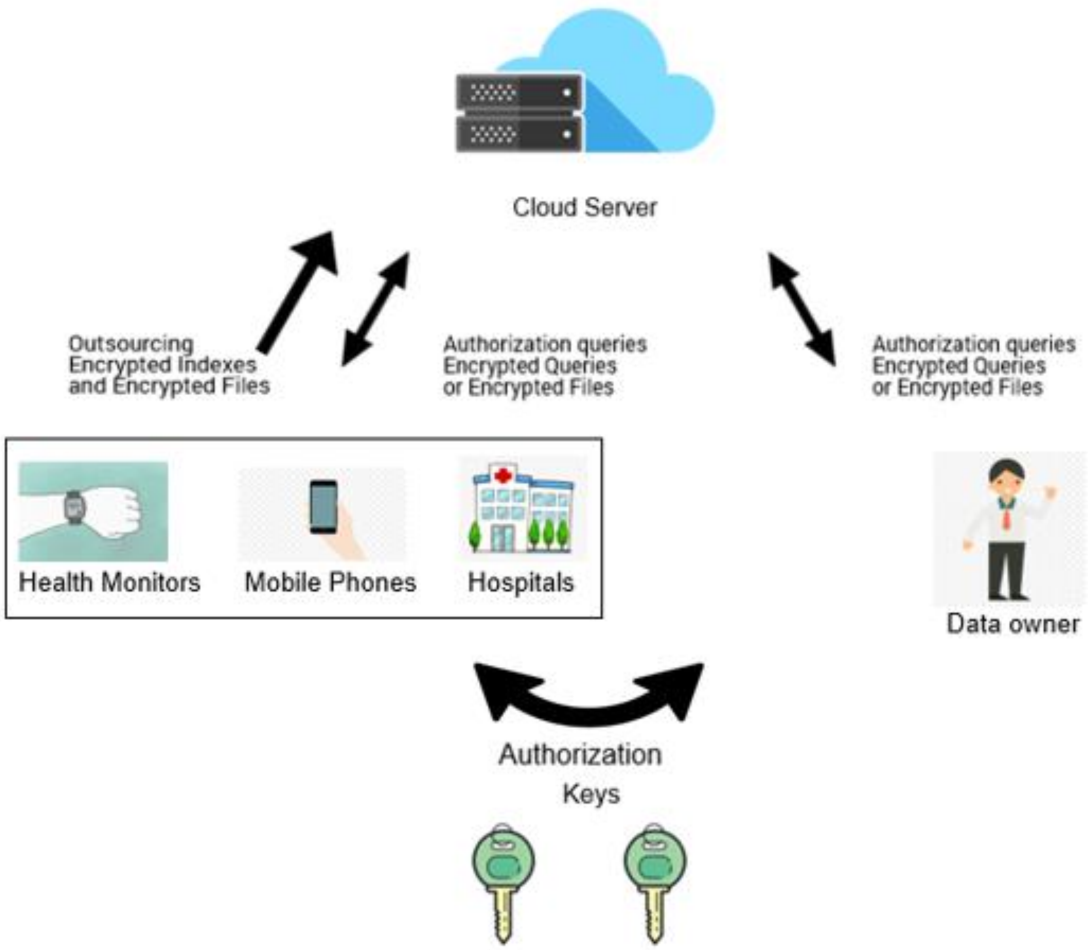

Figure 1: Example of Health Record Indexing

3. HEALTH RECORD INDEX

The purpose of Section 3 is to provide details regarding the scope and the usage of the HR index. Two different scenarios will be introduced, in order to identify the difference of using and not using the HR index, in the context of an Emergency scenario where quick access to the EHR data of the citizen by the authorized personnel, could be considered as of vital importance. To this end, this specific section describes the design of the HR index, including its components and its interfaces, in order to conclude into high level examples of using the proposed HR index.

3.1. Health Record Index Scope

Today's digital environment is characterized by a huge number of devices that enable data generation, processing and exchange. The data are stored either locally (on each device) or remotely (typically on cloud environments). This case is also true in the field of electronic health reflecting the exchange of data between citizens - patients and healthcare professionals through relevant devices. After a deep research and study, it was observed that there is no standardized way to exchange EHR data between citizens and medical staff, since more research has been attributed to the exchange of data only among the health personnel. At the same time, research was conducted on data exchange protocols, and it was observed that there is no relevant data exchange protocol that can be easily implemented and re-used simultaneously, by multiple operating systems. However, based on specific research, it was observed that what is mostly missing is the option of the citizens to store their EHR data on cloud repositories, and having the ability to provide to their healthcare practitioners a mean of accessing this data without specific credentials and without providing them a direct access (i.e. link) to their cloud repository, for security reasons.

Hence, the scope of the HR index is to cover the aforementioned gap. In summary, it will provide an intermediate for informing the healthcare practitioners about the cloud location of the stored EHR data without directly providing the data to them. The latter will facilitate emergency cases where the citizens are not able to provide their consent towards third-parties accessing their EHR data (supposing that in the past such a consent has been already approved by the citizen for granting access to the stored data with pre-specified data access tokens).

Based on the research conducted in Section 2, what is missing is that the proposed architectures are techniques that are not easily customizable and implementable in multiple domains and scenarios, since they are tailored to work under specific circumstances. Hence, what is needed and what the proposed HR index adds, is a standardized and easily implementable way in order to immediately access healthcare data stored on private clouds for further usage, through the indexing process.

3.2. Involved Applications

This section describes and lists the involved applications from the side of the S-EHR app and the HCP app.

3.2.1. S-EHR application

A S-EHR application (S-EHR app) is any application installed on a personal mobile device, that is able to store the personal health data of a user in a secure (encrypted) way according to the constraints specified by [D3.3] and that supports the InteropEHRate protocols defined in [D4.2]. Different vendors may develop different S-EHRs apps. A S-EHR app contains health data of the user, produced and signed (for traceability and trustability) by the healthcare organization that produces them, but can also contain data directly stored and produced by citizens or by sensors. The provenance and author of each health data is unambiguously persisted (i.e. “stored”) on the S-EHR and the principles of integrity and non repudiation are guaranteed. More details on what is supported by a S-EHR app can be found in [D2.5].

3.2.2. HCP application

An HCP application (HCP app) is a software application designed to provide medical staff with the ability to access and manage patients’ data from S-EHR apps, S-EHR Cloud and EHR of the Healthcare Organization. In other words, the HCP app is an application used by the HCPs to securely exchange health data of their EHRs with any S-EHR app and to read health data stored in S-EHR Cloud using the InteropEHRate protocols. More details on what is supported by an HCP app can be found in [D2.5].

3.3. Health Record Index Usage

The HR index is intended to be used in Emergency cases, where the citizen is not able to provide her consent for third-party accessing of her EHR data, supposing that in the past such a consent has been already approved by the citizen for granting access to the stored data to healthcare practitioners with pre-specified data access tokens. The following two subsections are depicting a reference scenario (i.e. Emergency Scenario as described in [D2.2]), where firstly (Section 3.3.1) the HR index is out of the context of the scenario, whereas in the second case (Section 3.3.2) the HR index is having a leading role in the overall scenario description.

3.3.1. Scenario without using the Health Record Index

As stated also in the previous sections, the purpose of the HR Index is to facilitate the use of the S-EHR cloud service during emergency situations. In order to highlight the need of the HR Index, in this and the following section a high-level explanation of the emergency scenario is given. What is described here, concerns the process that is followed when the HR Index is not present, while in the section below, the HR Index is introduced.

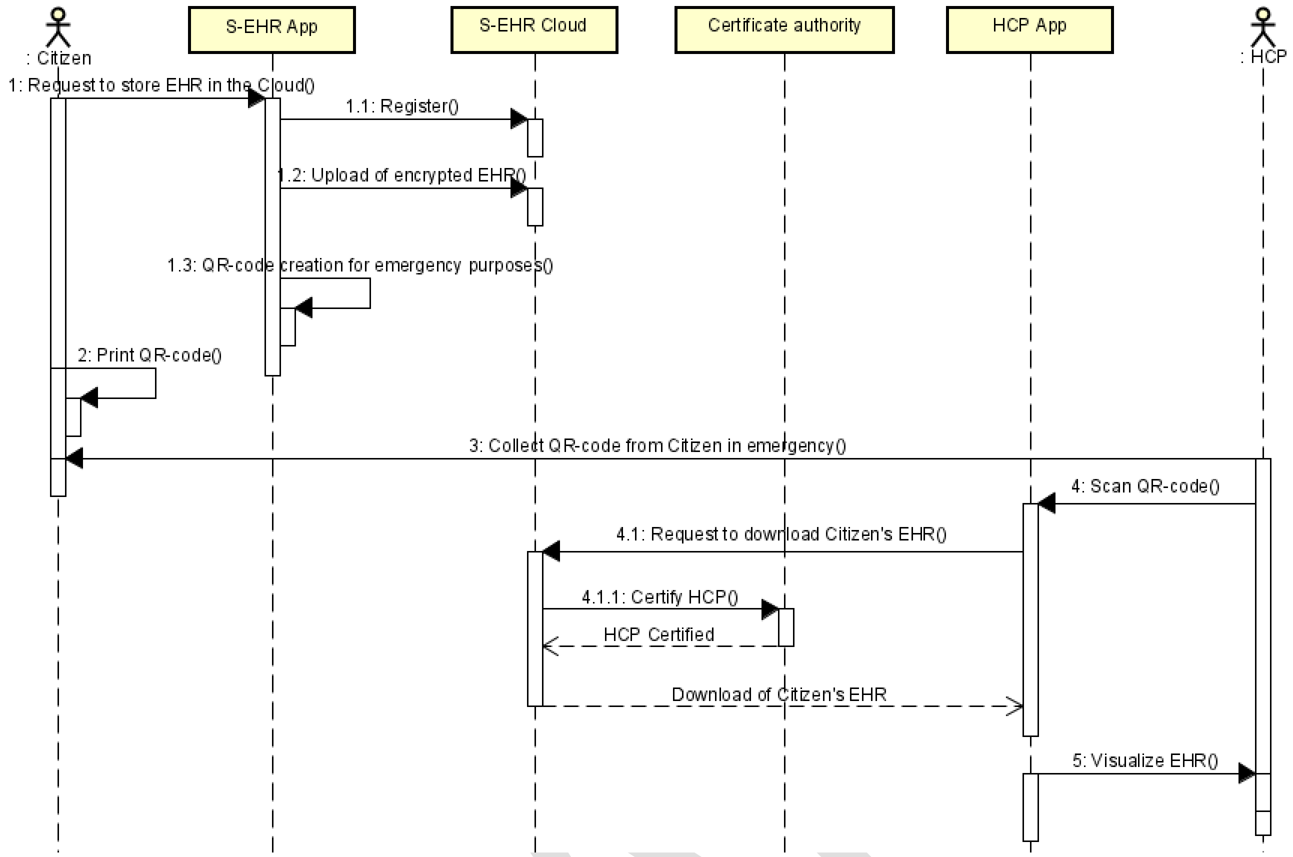

Figure 2: Emergency Scenario without using the Health Record Index

Prior to an emergency situation, a citizen may choose to use the optional service of the S-EHR cloud provided by InteropEHRate. This service's scope is two-fold; to back up the Citizen's EHR in the cloud, and to allow authorized HCPs to gain access to a Citizen's EHR when the Citizen's smart device is for some reason unreachable. If the Citizen agrees to these terms, a QR-code is created that contains information that will allow an HCP to access the S-EHR Cloud of the Citizen in need when in emergency. More specifically, what is stored in the QR-code includes a token that will be exploited by the S-EHR Cloud in order to grant access to the HCP to the Citizen's data, the symmetric key to decrypt the Citizen's EHR in the HCP app, and either S-EHR Cloud location, or the HR Index location if the first one is not available. This QR-code should always be printed by the Citizen and be carried along with her. When an emergency occurs and the Citizen's smart device is unreachable, the HCP scans the above-mentioned QR-code with the HCP app. Through the HCP app, the request to download the Citizen's encrypted EHR is forwarded to the S-EHR cloud service used by the Citizen. The S-EHR cloud grants access to the HCP as soon as she is authorized by the certificate authority. The encrypted EHR is downloaded and decrypted, for the HCP to visualize it. Figure 2 depicts the aforementioned emergency case, in high level details, since explaining the usage of each functionality is out of scope regarding the current deliverable.

As it was made clear, in this sub-scenario, the HR Index was not used. In several cases, this may lead to a few issues during an emergency that can be bypassed using the HR Index. One example of these issues is the case where the Citizen's preferred S-EHR cloud service is switched. In more detail, in the case that the Citizen decides to use another S-EHR cloud service, but does not have the time to print the QR-code with the new information before an emergency happens, the HCP in charge will not be able to access the EHR.

This problem is solved with the introduction of the HR Index, since it will be responsible to hold the information related to the Citizen's preferred cloud. In this case the QR-code will not contain information about the S-EHR cloud used by the Citizen, but information used by the HR Index instead. The HCP should in that case scan the QR-code, which will forward them to the HR Index and through that finally collect the address of the correct S-EHR cloud in order to download the Citizen's EHR.

3.3.2. Scenario using the Health Record Index

In this Section, the same scenario will be depicted, but it will also include the usage of the HR index. It should be noted that the HR index per se, should be considered as a database consisting of a table called USERS where it stores the unique ID of each citizen [column:ID]. When a new citizen ID is sent from the S-EHR cloud to the HR index, a unique table called ID_CLOUD_LOCATION is created in the HR index that has two columns [column: ID, column: location]. In the location column, it is stored the address of the cloud where the specific citizen has stored her data. If the citizen already exists, then the ID_CLOUD_LOCATION table is just updated with the new address of the cloud. Moreover, in order for the HR index to be correctly used, some preconditions must exist, as described below:

- The citizen should have a unique ID (e.g. a pseudo-identifier that is stored on the created QR code avoiding the usage of an identifier that can identify the citizen)
- The citizen should agree to store her EHR data on the S-EHR cloud
- When the storage of the EHR data to the S-EHR cloud happens, the S-EHR cloud sends to the HR index the unique ID of the citizen and the address of the cloud where the data is stored

Following the scenario described in Section 3.3.1, the additions to the scenario are that the citizen will have along with her the aforementioned QR code, including also her personal ID and the address of the HR index. When an emergency occurs, in the case that the Citizen's smart device is unreachable, the HCP scans the above-mentioned QR-code with the HCP app. This scanning redirects the HCP app to the HR index, locating the user from the USERS table that locates all the cloud addresses that are stored in the ID_CLOUD_LOCATION table. For the specific citizen, the HR index returns the cloud address, where a separate connection takes place to the cloud of the user, the EHR data is being retrieved and is finally provided to the HCP app. As in the previous case, the encrypted EHR is downloaded and decrypted, for the HCP to visualize it. Consequently, it has become clear that in the case that the Citizen decides to use another S-EHR cloud service, but does not have the time to print the QR-code with the new information before an emergency happens, the HCP is able to access the EHR data, since the HR index is having a dynamic behaviour, compared to the static behaviour of the QR code. Figure 3 depicts the aforementioned emergency case, in high level details, since explaining the usage of each functionality - apart from the usage of the HR index - is out of scope regarding the current deliverable.

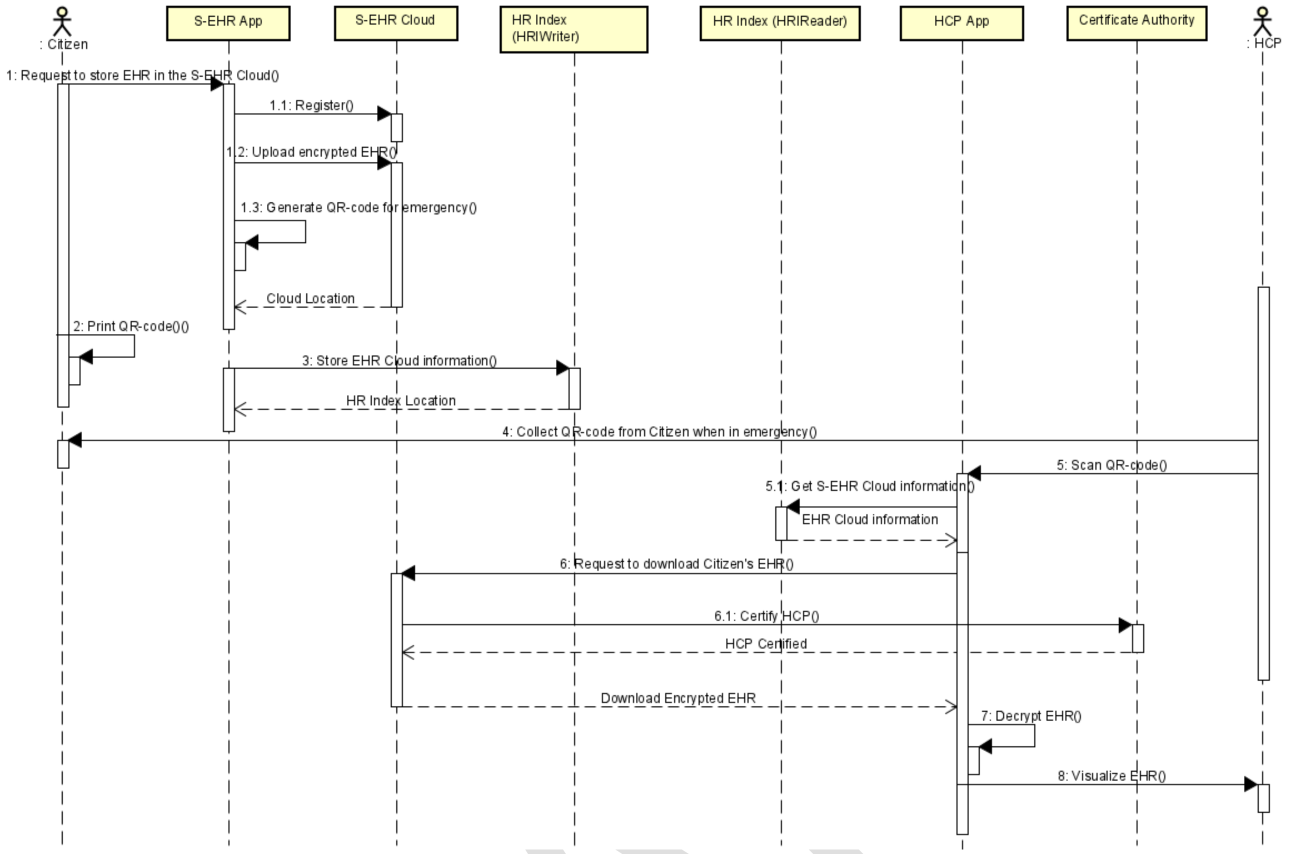

Figure 3: Emergency Scenario without the Health Record Index

3.4. Design of Health Record Index

It has been decided that the HR index will be provided as a single service, from the EU authority. In more detail, there will be a single HR index, in which every Healthcare Organization will have access to (after registering to it), and as a result the HCP that belongs to this organization can download EHR data through the cloud address located in the HR index. In the other case, where there would be a HR index at national-private level per healthcare organization, then there should be multiple HR indexes that would not be connected to each other. As a result, since the citizens' health records would be stored in different locations, the HR indexes would be almost impossible to interconnect to each other and be informed about the cloud locations where the citizens' data would be stored. Nevertheless, since establishing a European index is a complex process, such a case will be examined and addressed in the context of Work Package 8, during the exploitation phase, as well as the creation of the governance model phase.

In this context, in order for the HR index to be more easily implementable, in this deliverable it is proposed to design two Java libraries. The first library will be designed from the side of the S-EHR app, in the form of a Java-based component that can be nested in any Android application (Android Version 4.3 or higher). It offers a set of Java operations for uploading data to the cloud storage, using the HR index. The second library will be designed from the side of the HCP app, in the form of a Java-based component that can be embedded in any Java application (Web or Desktop applications). It will offer a set of Java operations enabling the application (used by a HCP) to download the data of a citizen from the cloud, using the HR index functionalities.

3.4.1. Design from the S-EHR-side

Regarding the S-EHR-side HR index library (i.e. M-HRI), this will contain all the operations that will be needed from the side of the S-EHR app developer to interact with the library and finally with the HCP app. This library will contain different operations that will have to be invoked in a specific sequence for implementing the purposes of the HR index, regarding the S-EHR app. As described previously, this library is a Java-based component that can be nested in any Android application. It offers a small set of Java operations for storing the cloud location to the HR index, and receiving back this location, in the context of the HR index.

3.4.1.1. Components

The M-HRI library incorporates a set of components (Figure 4) offering different functionalities and capabilities to the developer. These components can be offered publicly (i.e. Public components), including a single major component category: the Mobile HR index that includes all the operations and functionalities related to getting the indexing location. The Mobile HR index includes a single additional component category, namely Index Location, which is related to the functionalities for storing the cloud location to the HR index.

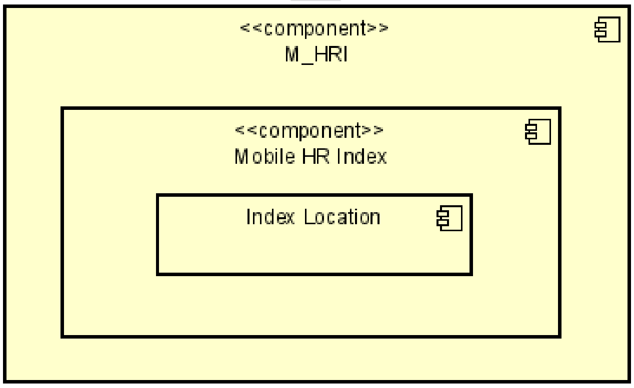

Figure 4: M-HRI Public Java Components

3.4.1.2. Remote interfaces of the M-HRI library

The following tables describe the remote interfaces of the M-HRI library that implement the communication between the S-EHR App and the Health Record Index.

Operation register

Name	register
Description	The citizen registers to the HR Index. The operation creates a new entry, containing the user's personal ID and the selected cloud's URL.
Arguments	• **citizenID**: A unique string that corresponds to the user using the HR Index • **cloudURL**: A string that contains the URL of the cloud that is used by the citizen to store their data.

Return Value	• **Response message** declaring that the user has been registered. • **Token**

Table 1 - Description of the register Operation as a remote interface

Operation getData

Name	getData
Description	This operation retrieves the data of a citizen stored in the HR Index.
Arguments	• **auth_token:** A unique token used to grant access to the citizen if they are authorized.
Return Value	• **Citizen's data** including the citizen's ID and the URL of the cloud where the data is stored

Table 2 - Description of the getData Operation as a remote interface

Operation updateData

Name	updateData
Description	This operation updates the information stored in the HR Index. In case the citizen changes the cloud used to store the data, the corresponding entry to the HR Index will be updated with new information.
Arguments	• **citizenID:** The citizen's unique ID created by their registration. • **cloudURL:** A string that contains the new URL where the data is now stored. • **auth_token:** A unique token used to grant access to the citizen if they are authorized.
Return Value	• **Response message** declaring that the data in the HR Index has been successfully updated.

Table 3 - Description of the updateData Operation as a remote interface

Operation removeData

Name	removeData
Description	This operation removes an entry from the HR Index in case a citizen no longer wants to give access to the data.
Arguments	• **citizenID:** The citizen's unique ID created by their registration. • **auth_token:** A unique token used to grant access to the citizen if they are authorized.
Return Value	• **Response message** declaring that the data in the HR Index has been successfully removed.

Table 4 - Description of the removeData Operation as a remote interface

3.4.1.3. Public Interfaces

The components defined in Section 3.4.1.1 are offering a specific interface, as depicted in Figure 5 and explained below.

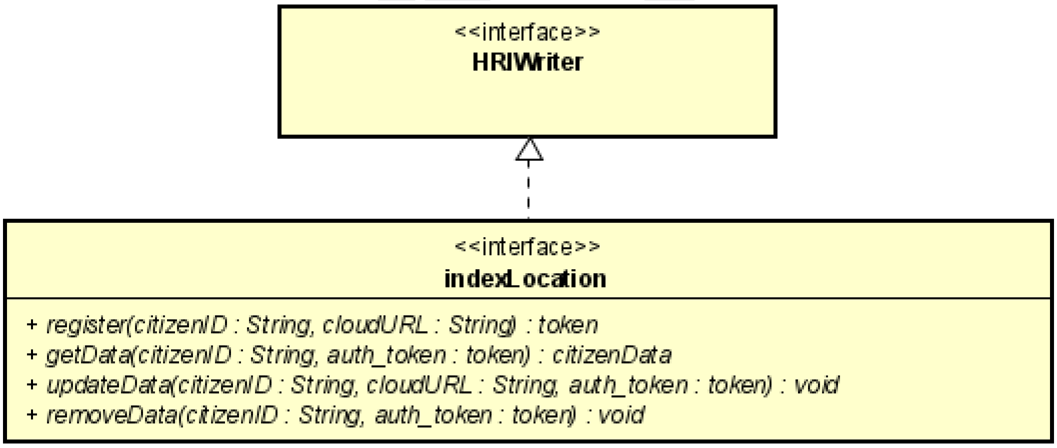

Figure 5: M-HRI Public Java Components Interfaces

HRIWriter

HRIWriter is the name of the interface that is offered by the Mobile HR Index component, containing the operations for letting the S-EHR app interact with the M_HRI library and finally receive the HR index location, depicting the cloud location where the EHR data is stored. HRIWriter is an interface containing a single additional interface for facilitating this communication process. This interface will be the indexLocation interface that contains a set of operations that have to be invoked for registering the user to the HR index, or for storing the cloud location where the EHR data of the citizen is stored and finally receiving back the according HR index location, regarding the side of the M-HRI library, among others. These operations are defined below.

Operation register

Name	register
URL	http://[base url]/citizen/register
HTTP method	POST
Description	The citizen registers to the HR Index. The operation creates a new entry, containing the user's personal ID and the selected cloud's URL.
Arguments	● **citizenID**: A unique string that corresponds to the user using the HR Index ● **cloudURL**: A string that contains the URL of the cloud that is used by the citizen to store their data.
Return Value	● **Response message** declaring that the user has been registered ● **Token**
HTTP return codes	**200** Successful: request was successfully processed. **400** Bad Request: search could not be processed or failed basic FHIR validation rules. **403** Forbidden: client is not allowed to access requested resources due to security policy. **409** Conflict: A citizen with the same id already exists. **500** Internal Server Error: server encountered an unexpected internal error, the request could not be processed.
Exceptions	● Security exceptions related to Cloud location. ● Network exceptions related to Internet connection.
Preconditions	● The S-EHR app has already stored the EHR data of the citizen to the cloud, and has received the specific cloud location

Table 5 - Description of the register Operation

Operation getData

Name	getData
URL	http://[base url]/citizen/getData
HTTP method	GET
Description	This operation retrieves the data of a citizen stored in the HR Index.
Arguments	• **auth_token:** A unique token used to grant access to the citizen if they are authorized.
Return Value	• **Citizen's ID** • **URL of the citizen's S-EHR Cloud**
HTTP return codes	**200** Successful: request was successfully processed. **400** Bad Request: search could not be processed or failed basic FHIR validation rules. **403** Forbidden: client is not allowed to access requested resources due to security policy. **409** Conflict: No citizen found with this ID. **500** Internal Server Error: server encountered an unexpected internal error, the request could not be processed.
Exceptions	• Security exceptions related to Cloud location. • Network exceptions related to Internet connection.
Preconditions	• The S-EHR app has already stored the EHR data of the citizen to the cloud, and has received the specific cloud location

Table 6 - Description of the getData Operation

Operation updateData

Name	updateData
URL	http://[base url]/citizen/updateData
HTTP method	PUT

Description	This operation updates the information stored in the HR Index. In case the citizen changes the cloud used to store the data, the corresponding entry to the HR Index will be updated with new information.
Arguments	● **citizenID:** The citizen's unique ID created by their registration. ● **cloudURL:** A string that contains the new URL where the data is now stored. ● **auth_token:** A unique token used to grant access to the citizen if they are authorized.
Return Value	● **Response message** declaring that the data in the HR Index has been successfully updated.
HTTP return codes	**200** Successful: request was successfully processed. bBad Request: search could not be processed or failed basic FHIR validation rules. **403** Forbidden: client is not allowed to access requested resources due to security policy. **409** Conflict: No citizen found with this ID. **500** Internal Server Error: server encountered an unexpected internal error, the request could not be processed.
Exceptions	● Security exceptions related to Cloud location. ● Network exceptions related to Internet connection.
Preconditions	● The S-EHR app has already stored the EHR data of the citizen to the cloud, and has received the specific cloud location

Table 7 - Description of the updateData Operation

Operation removeData

Name	removeData
URL	http://[base url]/citizen/removeData
HTTP method	DELETE
Description	This operation removes an entry from the HR Index in case a citizen no longer wants to give access to the data.
Arguments	● **citizenID:** The citizen's unique ID created by their registration. ● **auth_token:** A unique token used to grant access to the citizen if they are authorized.

Return Value	• **Response message** declaring that the data in the HR Index has been successfully removed.
HTTP return codes	**200** Successful: request was successfully processed. **400** Bad Request: search could not be processed or failed basic FHIR validation rules. **403** Forbidden: client is not allowed to access requested resources due to security policy. **409** Conflict: No citizen found with this ID. **500** Internal Server Error: server encountered an unexpected internal error, the request could not be processed.
Exceptions	• Security exceptions related to Cloud location. • Network exceptions related to Internet connection.
Preconditions	• The S-EHR app has already stored the EHR data of the citizen to the cloud, and has received the specific cloud location

Table 8 - Description of the removeData Operation

3.4.1.4. Usage of M_HRI

The following sequence diagram (Figure 6) shows the fundamental steps executed by the S-EHR app in order to firstly store the EHR data of the citizen to the cloud, and then store the cloud location where the EHR data of the citizen is stored and finally receive back the according HR index location. It should be noted that the storage of the EHR data to the cloud is out of context of this deliverable, hence this is provided in high level details. The usage of the defined operations is defined in the sequence diagram. Moreover, it is important to state that the following sequence does not show the real complexity and the complete interactions between components, because its main objective is to focus on interfaces, methods and data used by the S-EHR app.

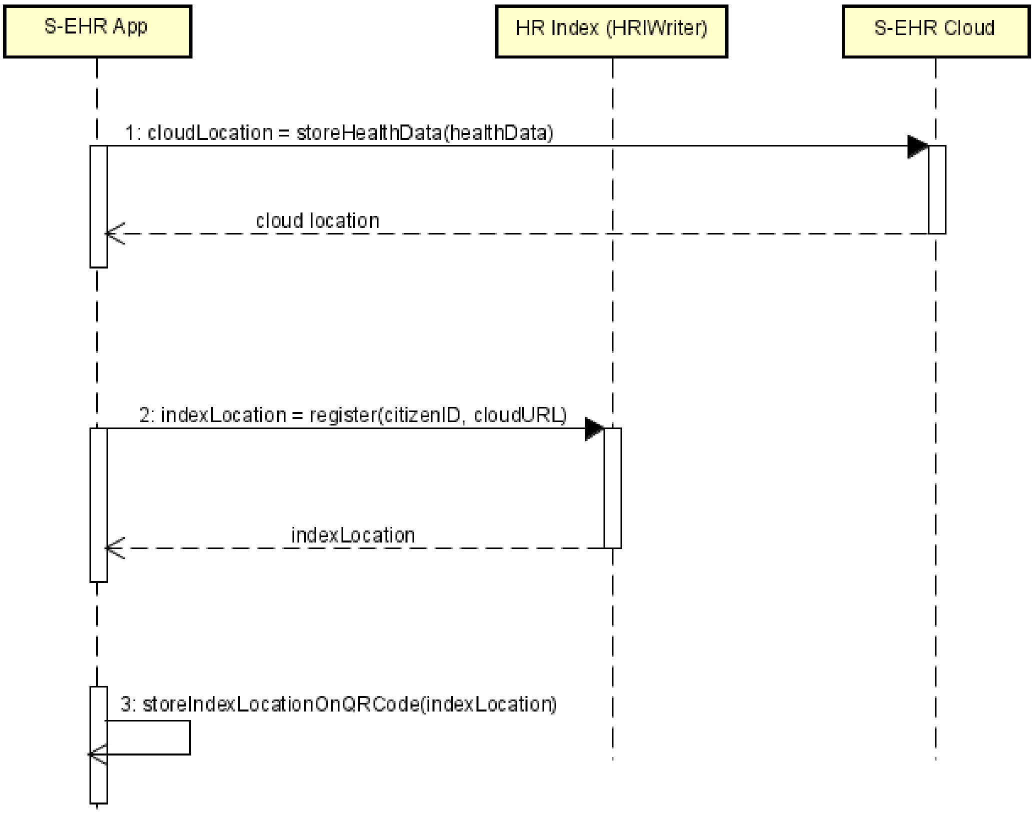

Figure 6: Example of storing the S-EHR Cloud location to the HR index

Step 1: The S-EHR app is uploading the EHR data to the S-EHR cloud.

Step 2: The S-EHR app stores the cloud location where the EHR data of the citizen is stored - along with a unique citizen identifier - and finally receives back the corresponding HR index location.

Step 3: The S-EHR app stores the received HR index location to the QR code, in order to be scanned in the future by any HCP, in case of emergency situations (as described in Section 3.3.2).

3.4.2. Design from the HCP-side

Regarding the HCP-side HR index library (i.e. T-HRI), this will contain the total of the operations that will be needed from the side of the HCP app developer to interact with the library and finally with the S-EHR app. This library will contain different operations that will have to be invoked in a specific sequence for implementing the purposes of the HR index, regarding the HCP app. As described above, the second library is a Java based component that can be embedded in any Java based application. It offers a small set of Java operations for receiving, from the HR index, the cloud location where the citizens' data is stored.

3.4.2.1. Components

The T-HRI library incorporates a set of components (Figure 7) offering different functionalities and capabilities to the developer. These components can be offered publicly (i.e. Public components), including a single major component category: the Terminal HR index that includes all the operations and functionalities related to getting the indexing location. The Terminal HR index includes a single additional component category, namely Cloud Location, which is related to the functionalities for getting the cloud location from the HR index.

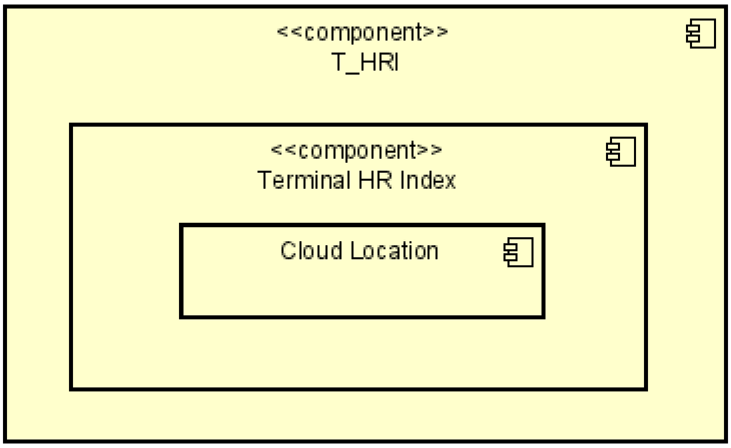

Figure 7: T-HRI Public Java Components

3.4.2.2. Remote interfaces of the T-HRI library

The following table describes the remote interfaces of the T-HRI library that implement the communication between the HCP Web App and the Health Record Index.

Operation getCloud

Name	getCloud
Description	This operation returns the cloud URL of the requested citizen to the HCP.
Arguments	• **citizenID:** The citizen's unique ID created by their registration.

Return Value	• **cloudURL:** The cloud address where the specific user stores their data.

Table 9 - Description of the getCloud Operation as a remote interface

3.4.2.3. HCP-side Public Interfaces

The components defined in Section 3.4.2.1 are offering a specific interface, as depicted in Figure 8 and explained below.

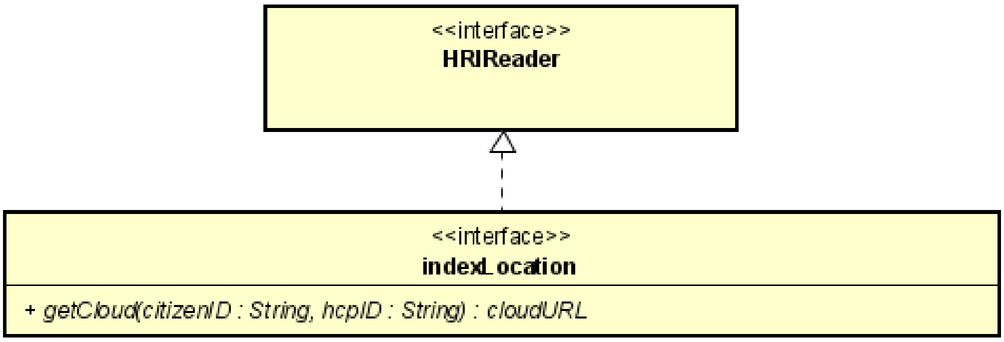

Figure 8: T-HRI Public Java Components Interfaces

HRIReader

HRIReader is the name of the interface that is offered by the Terminal HR Index component, containing the operations for letting the HCP app interact with the T_HRI library and finally receive the cloud location from the HR index, where the EHR data is stored. HRIReader is an interface containing a single additional interface for facilitating this communication process. This interface will be the indexLocation interface that contains the single operation that has to be invoked for getting the cloud location from the HR index, where the EHR data of the citizen is stored, regarding the side of the T-HRI library.

Operation getCloud

Name	getCloud
URL	http://[base url]/hcp/getCloud
HTTP method	GET
Description	This operation returns the cloud URL of the requested citizen to the HCP.
Arguments	• **citizenID**: The citizen's unique ID created by their registration.
Return Value	• **cloudURL**: The cloud address where the specific user stores their data.
HTTP return codes	**200** Successful: request was successfully processed. **400** Bad Request: search could not be processed or failed basic FHIR validation

	rules. **403** Forbidden: client is not allowed to access requested resources due to security policy. **409** Conflict: No citizen found with this ID. **500** Internal Server Error: server encountered an unexpected internal error, the request could not be processed.
Exceptions	● Security exceptions related to Cloud location. ● Network exceptions related to Internet connection.
Preconditions	● The S-EHR app has already stored the EHR data of the citizen to the cloud, and has received the specific cloud location ● A consent has been already approved by the citizen for granting access to the stored EHR data to HCPs with pre-specified data access tokens

Table 10 - Description of the getCloud Operation

3.4.2.4. Usage of T_HRI

The following sequence diagram (Figure 9) shows the fundamental steps executed by the HCP app in order to firstly identify the HR index location that depicts the cloud location where the EHR data of the citizen is stored, and finally facilitate the HCP to access this EHR data. It should be noted that the download of the EHR data from the cloud is out of context of this deliverable, hence this is provided in high level details. The usage of the operation getCloudLocation() is defined in the sequence diagram. Moreover, it is important to state that the following sequence does not show the real complexity and the complete interactions between components, because its main objective is to focus on interfaces, methods and data used by the HCP app.

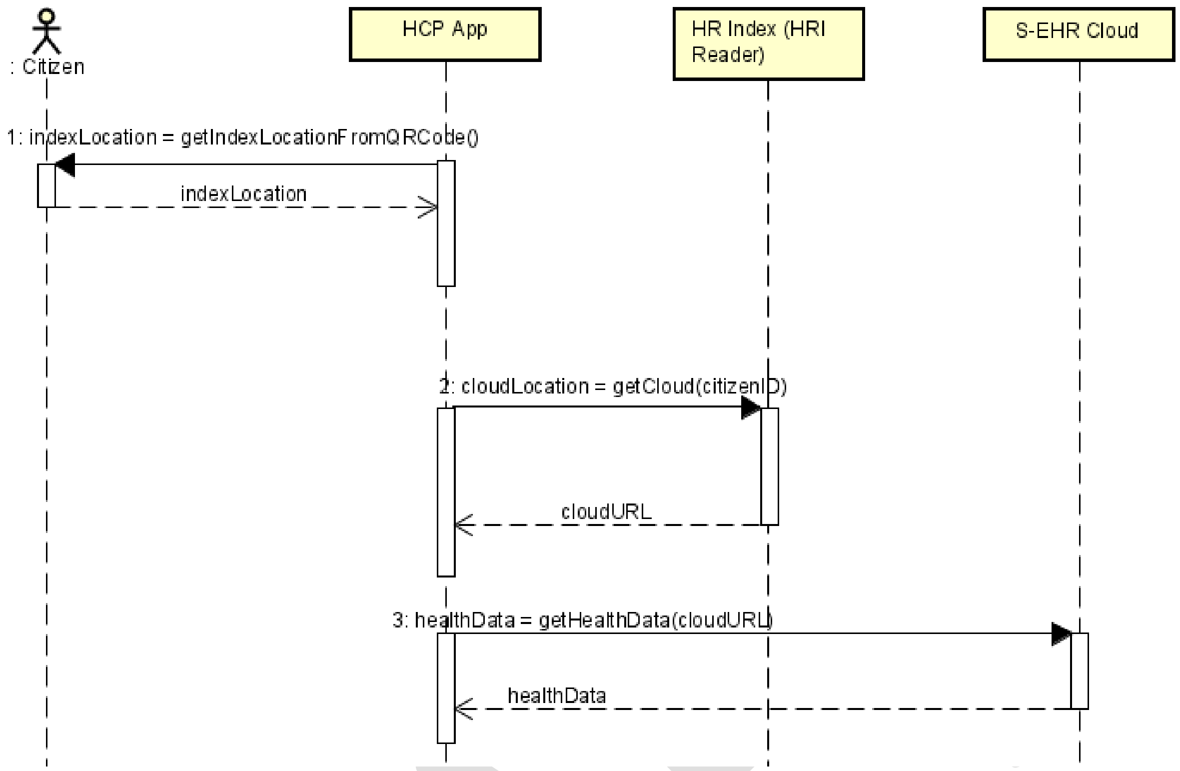

Figure 9: Example of downloading EHR data from the S-EHR Cloud location through the HR index

Step 1: The HCP app scans the QR code from the side of the S-EHR app, in order to receive the HR index location.

Step 2: The HCP app uses the HR index location - along with a unique citizen identifier - and receives back the corresponding cloud location, where the citizen's data is stored.

Step 3: The HCP app downloads the healthcare data of the citizen since the cloud location has been identified, and as a result the HCP visualizes this information to the HCP app (as described in Section 3.3.2).

4. CONCLUSIONS AND NEXT STEPS

The purpose of this deliverable was to propose an intermediate in the form of a HR index, for informing the healthcare practitioners about the cloud location, where citizens have uploaded their EHR data. The reason for that was to provide the HCPs the ability to access the stored EHR data without having direct access to this data, in order to address emergency cases where a citizen is not able to provide her consent for third-party accessing of this EHR data (supposing that in the past such a consent has been already approved by the citizen for granting access to the stored data to HCPs with pre-specified data access tokens). The research that has been conducted prior to specifying and designing the proposed HR index has been provided, including a detailed study of data and health data indexing techniques and methodologies. The difficulties and weak points of these techniques have been specified, concluding in end-to-end examples for making the purposes of the data and EHR data indexing clearly understandable. Afterwards, the overall vision, scope and usage of the HR index was provided. Two different scenarios were introduced, in order to identify the difference of using and not using the HR index, in the context of an Emergency scenario where quick access to the EHR data of the citizen by the authorized personnel, could be considered as of vital importance. Finally, the design of the HR index has been defined, including its components and its interfaces, in order to conclude into high level examples of using the proposed HR index.

To this end, this document presents the intended content of the HR index and its further functionality purposes. Following this document, a demonstration of this report is planned to be released in March 2022 (Deliverable D4.15), including the relevant implementation of the HR index, with additional details about the implementation code and its functionality.

REFERENCES

- **[D2.2]** InteropEHRate Consortium, *D2.2-User Requirements for cross-border HR integration - V2*, 2020. https://www.interopehrate.eu/resources/#dels

- **[D2.5]** InteropEHRate Consortium, *D2.5 - InteropEHRate Architecture - V2*, 2020. https://www.interopehrate.eu/resources/#dels

- **[D3.3]** InteropEHRate Consortium, *D3.3-Specification of remote and D2D IDM mechanisms for HRs Interoperability - V1*, 2019. https://www.interopehrate.eu/resources/#dels

- **[D4.2]** InteropEHRate Consortium, D4.2-Specification of remote and D2D protocol and APIs for HR exchange - V2, 2020. https://www.interopehrate.eu/resources/#dels

- **[EHRLER 2007]** Ehrler, Frédéric, et al. "Challenges and methodology for indexing the computerized patient record." Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems. IOS Press, 2007.

- **[PEREIRA 2008]** Pereira, Suzanne, et al. "Using multi-terminology indexing for the assignment of MeSH descriptors to health resources in a French online catalogue." AMIA Annual Symposium Proceedings. Vol. 2008. American Medical Informatics Association, 2008.

- **[WAN 2019]** Wan, Shaohua, et al. "Multi-dimensional data indexing and range query processing via Voronoi diagram for internet of things." Future Generation Computer Systems 91 (2019): 382-391.

- **[YAO 2018]** Yao, Xin, et al. "Privacy-preserving search over encrypted personal health record in multi-source cloud." IEEE Access 6 (2018): 3809-3823.

- **[YUAN 2016]** Yuan, Xingliang, et al. "Enabling secure and fast indexing for privacy-assured healthcare monitoring via compressive sensing." IEEE Transactions on Multimedia 18.10 (2016): 2002-2014.