Indexing of Cloud Stored Electronic Health Records for Consented Third Party Accessing

Athanasios Kiourtis, Argyro Mavrogiorgou, Chrysostomos Symvoulidis, Charalampos Tsigkounis, Dimosthenis Kyriazis
University of Piraeus
Piraeus, Greece
{kiourtis, margy, simvoul, ctsigk, dimos}@unipi.gr

***Abstract*—Electronic Health Record (EHR) data is being required to be shaped into a standardized and interoperable format for easy and secure accessing from anywhere, towards the vision of a borderless Health Information Exchange (HIE). Towards this vision, EHR data is being securely stored in different cloud repositories, making it however difficult and time-consuming to access it in emergency cases from healthcare practitioners, or even from the owners. In order to access the EHR data, it is needed specific credentials and real-time approved consent, which cannot be feasible in cases where this data must be urgently accessed and furtherly processed. In this paper, a health record indexing methodology is being proposed (i.e. HR index) that facilitates the real-time accessing of non-privileged users, to the cloud stored EHR data of citizens who are facing emergency situations. 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. The overall vision, scope and usage of the HR index is provided in the context of an Emergency scenario, concluding to the design and evaluation of the HR index, under specific circumstances and requirements.**

I. INTRODUCTION

Electronic health (eHealth) can be best thought as an evolving domain combining medical informatics, public health, and business, using the health services and information distributed over the Internet and associated technologies [1]. This term includes technical progress which includes contributing and expanding health care locally, countywide, and universally by using information and communication technologies (ICT). Based on the World Health Organization (WHO) [2] the most common areas of eHealth are the facts of delivering health information, for health professionals and health consumers, through ICT and e-commerce to improve public health services, and using e-commerce and e-business practices in health systems management.

With eHealth surrounding every aspect of our daily routine, there have been seen an explosive health data growth rate of 878 percent since 2016, reaching 8.41 petabytes on average in 2018, according to statistics compiled by Dell EMC [3]. By 2025, global estimates suggest 463 exabytes of data will be created each day, where especially in healthcare, current estimates suggest that a single patient will generate close to 80 megabytes each year in imaging and electronic medical record (EMR) data [4]. Collecting and analyzing such data helps to get more insights about the people's health and understand the reason why certain health conditions occur. Moreover, the benefits of such amounts of data in healthcare can be translated into terms of timely and improved patient's treatment and care experience. Furthermore, prediction of deadly disease, reduction of death rates, increase of healthy lifestyle and effective surveillance of public health, and making the policies for public healthcare and management system, can be characterized among the advantages of using and analyzing the vast amounts of healthcare data [5].

However, even though nearly all hospitals and healthcare practitioners (HCPs) use electronic systems in their facilities, unless those providers choose to connect to a health network such as a health information exchange, that information is not efficiently and easily shared [6]. There may exist one or two connections to other providers, but the data is not shared with all providers who treat a specific patient. Furthermore, the data collected for all the public health statistics, like lab results, are not exchanged with all the patient's HCPs or sent towards sufficient health information exchange (HIE).

To understand how the HCPs access could be improved towards accurate and timely data, HIE should be described as a type of health care public utility. For instance, in the case that a citizen was admitted to a local hospital with a diagnosis of COVID-19, after her recovery and during her discharge, she would be instructed by the hospital staff to get another X-ray at the imaging center and schedule a follow up visit with her primary care provider. In the case that the hospital, imaging center, and primary care provider were all connected to a specific HIE system, the laboratory results, medications, and HCPs notes – including the aforementioned X-ray, would be easily accessible and available to any HCP. Such a result could improve the quality of care, ensure that medications are accurate, and reduce the chance that the HCP would need to repeat medical examinations, which could be time consuming and could increase costs. However, mainly due to state and federal regulations, as well as current public policies, most of the healthcare organization and practitioners are not able to perform sufficient HIE.

The importance of HIE [7] can be best described especially in emergency situations, where the exchange of data among citizens - patients and healthcare professionals, could even save someone's life. In this context, this could be performed through citizens storing their Electronic Health Record (EHR) data on cloud repositories and having the ability to provide to their HCP the means of accessing these EHR data without specific credentials [8]. Towards this goal, the purpose of this paper is to propose an innovative proxy in the form of a Health Record index (HR index), for informing the HCPs about the cloud location of the stored EHR data of the patients, 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-party access to 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 HCPs with pre-specified data access tokens). Consequently, the focus of this paper is to define the technical specifications of the proposed HR index, providing also detailed descriptions of its context of use, outlining its functionality, accompanied by an explanation of its purpose of existence.

The rest of this document is structured as follows. Section II includes the related work and research regarding the fields of data indexing and health data indexing. Moreover, Section III describes the proposed HR index and its applicability, explaining the design of two different programming libraries that implement the functionality of the HR index, offering a set of operations for uploading data to the cloud storage and downloading the data of a citizen from the cloud, using the proposed HR index. Section IV illustrates a preliminary evaluation and discussion of the current indexing technique, and Section V includes our conclusions, remarks, and future plans.

II. Related Work

A. Indexing in Databases

Indexing is a technique of data structures which is used in order to locate and access quickly the data that has been stored in a database [9]. Data indexing is an efficient way of processing queries and retrieving data from a database, as it minimizes the number of disk accesses [10]. An index looks like a table which has two columns. The first column represents a copy of the primary or candidate key of the database's table, while the second one represents the pointer. Each pointer corresponds to the address of the disk block where the data is stored. In contrast with tables, indexes are sorted, and this is how the query's efficiency is maximized. By using indexes, the time of processing and retrieving data is reduced and there is no need in accessing the entire table row by row [11]. When the table is unsorted, all records must be accessed linearly, which in the case of large databases or complex queries, is not very efficient as it may take a lot of time.

There exist several techniques of data indexing, including the primary, the secondary and the multilevel indexing.

1) Primary indexing: The primary indexing is a fixed-length data file that has two fields [12]. The first field corresponds to the table's primary key and the second one contains a pointer which corresponds to the address of the disk block where the data is stored. The primary index is sorted based on the primary key and there is a one-to-one relationship between the entries in the index and the records in the main table [13]. The primary index is of two types: the dense index, and the sparse index. In dense index, each record consists of a search key which is associated with a pointer that indicates the actual record on the disk [14]. The number of records in the data file is equivalent to the number of records in the index and both the search and the primary keys are in the same order [15]. This technique makes lookup faster, but it also requires more space to store the index's records and this might lead to an overhead in memory. Dense indexes can also support range queries. In that case, all records, which are located between the minimum and the maximum value, will be retrieved as follows. Firstly, the minimum value will be located and all the sequential blocks that have a key less or equal to the maximum value will be loaded also in memory. To prevent the above-mentioned overhead in memory, sparse indexes do not contain a key for each record in the data file, but they hold a key that points to each block of the data file instead. In this way, the required space in memory is less than before, but the time needed for retrieving a record based on a key is more. The process for retrieving a record is as follows. First of all, there will be a search in sparse index according to the search key, and if the data are not directly where the pointer indicates then the block address will be fetched linearly until the requested data is found and retrieved.

2) Secondary indexing: The secondary indexing, also known as non-clustering index, is also a data file that has two fields, the search key, and the pointer [16]. In contrast with the primary index, the search key does not have a one-to-one relation with the records in the main table. The search key is not a primary key, but a candidate key of the table and it may point to more than one block into disk. The search keys are also sorted, as opposed to the stored data which is neither sorted nor in the same order as in the index. Secondary indexes are used when a query is not performed based on the primary key and they also require more time compared to the primary indexes, since more procedures are needed to retrieve the data from the disk. In conclusion, in primary indexes, the sparse index consumes less space compared to dense index, which needs more space, but it is faster, and the computation cost is lower. Moreover, the secondary indexes are less efficient than the primary ones, since both space and retrieving time is larger.

3) Multilevel indexing: The multilevel indexing can be created by any of the abovementioned indexes, which are also known as single-level indexes, as long as they consist of more than one disk blocks [17]. Essentially, a multilevel index is created when the single-level index does not fit in memory and it is stored on the disk along with the database. In this method, it is important to keep the records of the index in memory in order to decrease the time needed for the search operation and to reduce the number of disk accesses. The multilevel structure is as follows. A single-level index is placed at the first level of the multilevel index and a primary index is built on the first level index and composes the second level of the multilevel index. Each entry of the second level index corresponds to a disk block of the first level index's entries and this process can be repeated for the second level index and so on, until all the entries of a level fit in a single block

B. Health Record Indexing

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.

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 paper. 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 [18] 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 a document should be retrieved. In the same context, the authors in [19] 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 [20] 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 [21] 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.

C. Advancements beyond the Related Work

Through the secure storage of the EHR data in different cloud repositories, the goal of HIE can become a reality, with increased benefits for both the HCPs and the citizens. However, the difficulty of accessing the EHR data in emergency situations from HCPs, or even from the citizens' themselves is being increased. Several techniques have been proposed and are currently under development, which are however based on the fact that each interacting party should have specific credentials for accessing this data in private cloud repositories, which can be almost impossible in urgent situations where this data must be accessed no matter what. Furthermore, what is also missing is that the proposed architectures are techniques 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 of immediately accessing healthcare data stored on private clouds for further usage, through a specific indexing process. Through the proposed methodology, it is being facilitated the access of non-privileged users to the EHR data stored in cloud repositories of citizens-in-need under emergency cases. These users will make use of specifically tailored applications, that will interact with each other in a standardized way, for achieving the required data accessing and further exploitation.

III. HEALTH RECORD INDEX

A. 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 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 HCPs 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 a proxy for informing the HCPs 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).

B. Involved Applications

In order for the HR index to become a reality, two different applications have to be developed and used by the citizens and the HCPs accordingly. The Smart-EHR (S-EHR) application and the HCP application are described below.

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 [22] and that supports the InteropEHRate protocols defined in [23]. Different vendors may develop different S-EHR 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. More details on what is supported by a S-EHR app can be found in [24].

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 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 cloud repositories, using the InteropEHRate protocols defined in [23]. More details on what is supported by an HCP app can be found in [24].

C. 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 HCPs with pre-specified data access tokens. The following two subsections are depicting a reference scenario (i.e., Emergency Scenario), where firstly (sub-Section 1) the HR index is out of the context of the scenario, whereas in the second case (sub-Section 2) the HR index is having a leading role in the overall description.

1) 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 data when the citizen's smart device is for some reason unreachable. In the case that the citizen wants to back up her EHR data to the cloud, she must request to store her data to the S-EHR app in order to register to the S-EHR cloud, and then upload the encrypted EHR data. If the citizen agrees to these terms, a Quick Read code (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. 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 data 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 data is downloaded and decrypted, for the HCP to visualize it. Fig. 1 depicts the aforementioned emergency case.

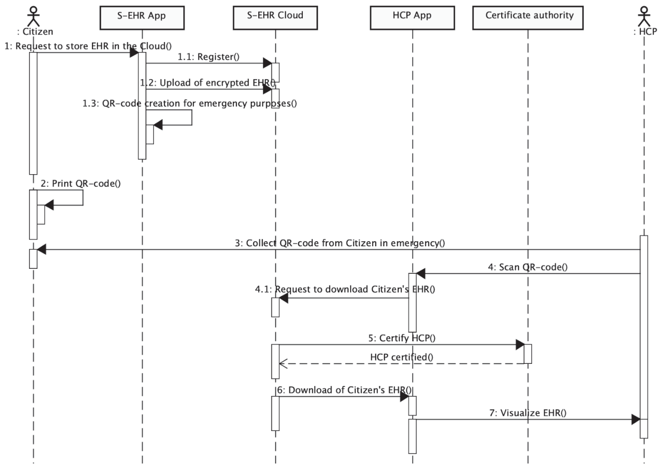

Fig. 1. Emergency Scenario without using the Health Record Index

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 data. 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 her 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 data.

2) Scenario using the Health Record Index

In this sub-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., Social Security Number)
- 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 which was already described, 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 different cloud addresses, where for each different cloud address 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 behavior, compared to the static behavior of the QR-code. Fig. 2 depicts the aforementioned emergency case.

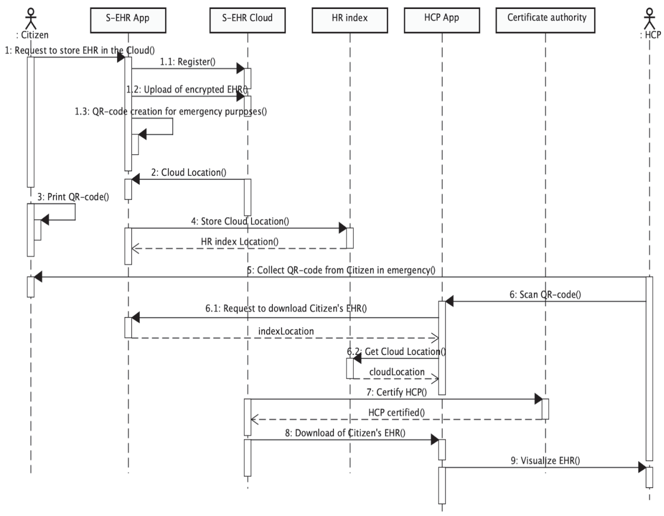

Fig. 2. Emergency Scenario with the Health Record Index

C. Design of the Health Record Index

Following a deep research, it has been decided that the HR index will be provided as a single service, from the European Union (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 single 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.

In this context, in order for the HR index to be more easily implementable, it is proposed to design two Java programming 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 will offer 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 an HCP) to download the data of a citizen from the cloud, using the HR index functionalities.

1) Design from the S-EHR app Side

Regarding the S-EHR app 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 (version 8 or higher). It will offer a 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.

The M-HRI library incorporates a set of components (Fig. 3) 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 for 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.

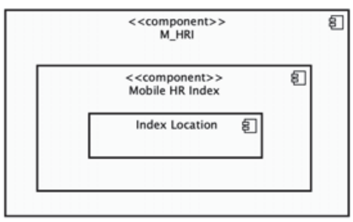

Fig. 3. M-HRI Public Java Components

The defined components are offering a specific interface, as depicted in Fig. 4 and are furtherly explained.

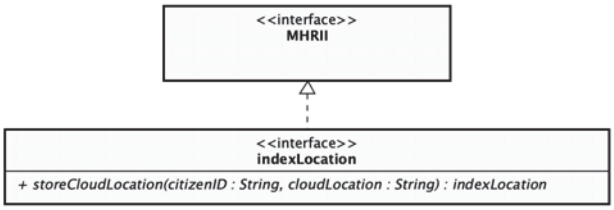

Fig. 4. M-HRI Public Java Components Interfaces

MHRII 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. MHRII 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 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. Table I explains the functionality and the identity of this single operation (i.e., storeCloudLocation).

TABLE I. OPERATION FOR STORING THE CLOUD LOCATION

Operation Details	Operation Explanation
Name	storeCloudLocation
Description	This operation is invoked by the S-EHR app for storing the cloud location where the EHR data of the citizen is stored and finally receiving back the corresponding HR index location.
Arguments	- citizendID: a unique string that identifies the citizen who is using the HR index functionality. - cloudLocation: a string that contains the URL of the cloud where the citizen has uploaded her EHR data.
Return Value	This operation will return the HR index location containing the location of the cloud where the citizen has stored her EHR data
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 cloud location

2) Design from the HCP app Side

Regarding the HCP app 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' EHR data is stored.

The T-HRI library incorporates a set of components (Fig. 5) 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.

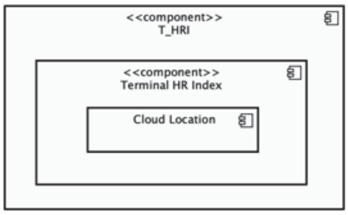

Fig. 5. T-HRI Public Java Components

The defined components are offering a specific interface, as depicted in Fig. 6 and are furtherly explained.

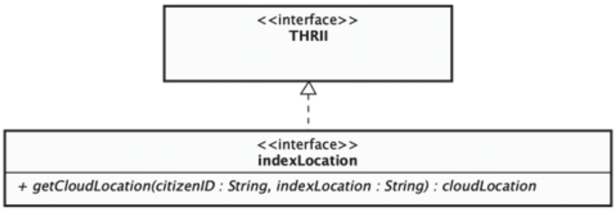

Fig. 6. T-HRI Public Java Components Interfaces

THRII 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. THRII 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. Table II explains the functionality and the identity of this single operation (i.e., getCloudLocation).

TABLE II. OPERATION FOR GETTING THE CLOUD LOCATION

Operation Details	Operation Explanation
Name	getCloudLocation
Description	This operation is invoked by the HCP app for getting the cloud location from the HR index, where the EHR data of the citizen is stored.
Arguments	- citizendID: a unique string that identifies the citizen who is using the HR index functionality. - indexLocation: a string that contains the specific location of the HR index and belongs to the citizen with the unique citizenID, who has previously uploaded her EHR data to the cloud.
Return Value	This operation will return the cloud location containing where the citizen has stored her EHR data
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 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

IV. Preliminary Evaluation

A. Working Environment

In order to evaluate the proposed HR indexing methodology, two applications were created in Java for Android, using Android Studio [25], for uploading and downloading healthcare data based on the described HR index flow. The cloud storage that was used was Dropbox [26] cloud-based storage system, while the dataset that was stored on the cloud repository and used for the evaluation purposes, included the demographic details of a random citizen (i.e., name, address, contacts, and ID) (Fig. 7), based on the HL7 FHIR format [27], as well as the allergies and intolerances of the specific citizen.

```xml
<?xml version="1.0" encoding="UTF-8"?>

<Patient xmlns="http://hl7.org/fhir">
  <id value="example"/>
  <text>
    <status value="generated"/>
    <div xmlns="http://www.w3.org/1999/xhtml">
      <table>
        <tbody>
          <tr>
            <td> Name</td>
            <td> Peter James
              <b> Chalmers</b>  (&quot;Jim&quot;)
            </td>
          </tr>
          <tr>
            <td> Address</td>
            <td> 534 Erewhon, Pleasantville, Vic, 3999</td>
          </tr>
          <tr>
            <td> Contacts</td>
            <td> Home: unknown. Work: (03) 5555 6473</td>
          </tr>
          <tr>
            <td> Id</td>
            <td> MRN: 12345 (Acme Healthcare)</td>
          </tr>
        </tbody>
      </table>
    </div>
  </text>
</Patient>
```

Fig. 7. Demographic details to be stored in HL7 FHIR format

B. Evaluation and Discussion of Results

The overall flow of Fig. 2 was followed, providing us with the expected results. Fig. 8 displays a few screenshots of the developed applications that confirm the functionality of the HR indexing methodology, showcasing (a) the scanning of a Quick-Read (QR) code for redirecting the HCP app to the HR index, locating the citizen and the stored cloud addresses, and (b) the successful connection to the HR index, while Fig. 9 depicts the final access of the HCP to the citizen's EHR data.

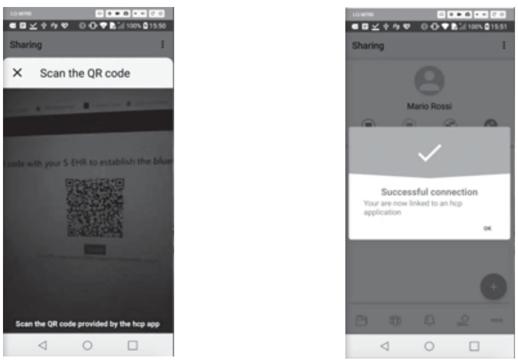

Fig. 8. Applications' user interfaces using the HR index

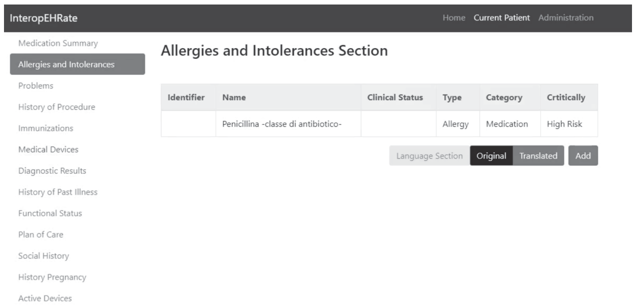

Fig. 9. Applications' user interfaces using the HR index

Based on the provided results it can be identified that the HR index could be defined as innovative proxy, for informing the HCPs about the cloud location, where citizens have uploaded their EHR data. All the involved parties were able to easily follow the proposed standardized process and gather the required results. It should be noted that additional testing was performed during the HR index evaluation with datasets of different sizes and multiple cloud-based storage systems (i.e., ownCloud [28], Google Drive [29]). More particularly, eight (8) different datasets of different overall sizes have been used (structured in HL7 FHIR format), while their total transmission time was calculated, as the total time that was needed to upload, access, and download the EHR data. Table III depicts the overall transmission times, for each case.

TABLE III. Overall transmission times for various cloud-based storage systems

Overall size (Kbytes)	Cloud-based Storage System	Transmission Time (sec)
8	Dropbox	3
3	Google Drive	2
18	ownCloud	5
172	Google Drive	8
339	Google Drive	11
708	ownCloud	13
1412	ownCloud	17
4008	Dropbox	29

The results were as expected, whereas there were some cases – in case of increased data sizes, where the uploading and downloading of the EHR data, was leading to increased waiting times (latency). However, this was due to bandwidth restrictions since the volume of information which had to be sent over the connection was increased. Moreover, the usage of different Cloud-based Storage Systems did not derive any measurable results, which could lead in identifying the most efficient storage system for the HR indexing purposes. For that purpose, the same experiments were repeated where each different dataset of Table III was stored in all the different Cloud-based Storage Systems, in order to repeat the HR indexing process and measure the total transmission time. Table IV depicts the overall transmission times, for each different case.

TABLE IV. OVERALL TRANSMISSION TIMES FOR EACH DIFFERENT CLOUD-BASED STORAGE SYSTEM

Overall size (Kbytes)	Transmission Time (sec)		
	Dropbox	*Google Drive*	*ownCloud*
8	3	3	3
3	2	2	2
18	5	6	5
172	8	8	8
339	10	11	11
708	12	14	13
1412	17	18	17
4008	29	28	28

As it was already predicted, the usage of different Cloud-based Storage Systems did not derive any measurable results, which could lead in identifying the most efficient storage system for the HR indexing purposes, since all of them behaved in a similar way for the different datasets. In any case, regardless of the Cloud-based Storage Systems used and the dataset sizes, the final outcomes were identical to the ones which are described in the evaluation scenario.

V. CONCLUSION

Current healthcare technologies promise to address the challenge of healthcare data exchange with the aim of helping automating efforts, increasing transparency, and reducing bad communication between health plans, providers, and healthcare organizations. It is undeniable that HIE can allow clinicians, nurses, pharmacists, and other healthcare providers and patients to access and securely share a patient's vital medical information electronically, improving the speed, quality, safety, and cost of patient care. While HIE cannot replace provider-patient communication, it can greatly improve the completeness of patients' records.

The purpose of this paper was to propose an innovative proxy in the form of a HR index, for informing the HCPs 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, to conclude into high level examples of using the proposed HR index.

To this end, it is within our next goals to perform additional evaluation of the proposed HR indexing methodology, with multiple datasets and Cloud-based Storage Systems, to conclude in domain-specific rules and constraints of following a pre-described process for already defined requirements. Moreover, future work concerns deeper analysis of particular mechanisms, including further investigation, implementation and evaluation, based on the current eHealth and Healthcare trends and needs, tackling security requirements, public policies and regulations. To this end, we will continue evaluating the proposed HR indexing mechanism taking into consideration datasets of multiple medical standards and formats, including formats of unknown nature, respecting privacy issues [30].

ACKNOWLEDGMENT

The research leading to this result has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106 (InteropEHRate project).

REFERENCES

[1] S. G. Alonso, J. Arambarri, M. López-Coronado, and I. la Torre Díez, "Proposing new blockchain challenges in ehealth. Journal of medical systems", vol.43(3), 2019.
[2] I. Khaleel, B. C. Wimmer, G. M. Peterson, S. T. R. Zaidi, E. Roehrer, E. Cummings, and K. Lee, "Health information overload among health consumers: A scoping review". Patient education and counseling, vol. 103(1), 2020, pp. 15-32.
[3] "How Healthcare Organizations Can Protect Data Into The Next Decade And Beyond". Website: https://www.forbes.com/sites/forbestechcouncil/2020/03/04/how-healthcare-organizations-can-protect-data-into-the-next-decade-and-beyond/
[4] "Better Patient Outcomes Through Mining of Biomedical Big Data". Website: https://www.frontiersin.org/articles/10.3389/fict.2018.00030/full
[5] A. Silverio, P. Cavallo, R. De Rosa, and G. Galasso, "Big health data and cardiovascular diseases: A challenge for research, an opportunity for clinical care". Frontiers in medicine, vol.6(36), 2019.
[6] L. Lenert, and B. Y. McSwain, "Balancing health privacy, health information exchange, and research in the context of the COVID-19 pandemic". Journal of the American Medical Informatics Association, vol.27(6), 2019, pp. 963-966.
[7] P. Esmaeilzadeh, T, Mirzaei, and M. Maddah, "The effects of data entry structure on patients' perceptions of information quality in Health Information Exchange (HIE)". International Journal of Medical Informatics, vol. 135, 2019.
[8] X. Liu, Z. Wang, C. Jin, F. Li, and G. Li, "A blockchain-based medical data sharing and protection scheme". IEEE Access, vol.7, 2019, pp. 118943-118953.
[9] "Indexing in Databases | Set 1". Website: https://www.geeksforgeeks.org/indexing-in-databases-set-1/
[10] "Indexing in Databases with EXAMPLES". Website: https://www.guru99.com/indexing-in-database.htm
[11] "How Does Indexing Work". Website: https://chartio.com/learn/databases/how-does-indexing-work/
[12] "Primary Indexing in Databases". Website: https://www.geeksforgeeks.org/primary-indexing-in-databases/
[13] "Indexing in DBMS". Website: https://www.javatpoint.com/indexing-in-dbms
[14] "DBMS – Indexing". Website: https://www.tutorialspoint.com/dbms/dbms_indexing.htm
[15] "Free University of Bolzano–Database 2. Lecture II, 2003/2004 – A. Artale". Website: https://www.inf.unibz.it/~artale/DB2/handout2.pdf
[16] "Introduction to Indexing in SQL. Website: https://www.datacamp.com/community/tutorials/introduction-indexing-sql

[17] “Index Structures for Files: Multi-Level Indexes”. Website: https://www.scss.tcd.ie/Owen.Conlan/4d2/4D2-9&10_Multi-Level_Indexes_v1.02.pdf
[18] F. Ehrler, 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, 2007.
[19] X. Yuan, et al., "Enabling secure and fast indexing for privacy-assured healthcare monitoring via compressive sensing”, IEEE Transactions on Multimedia, vol.18(10), 2016, pp. 2002-2014.
[20] S. Pereira, 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.
[21] S. Wan, et al., “Multi-dimensional data indexing and range query processing via Voronoi diagram for internet of things”. Future Generation Computer Systems, vol.91, 2019, pp. 382-391.
[22] InteropEHRate Consortium, D3.3-Specification of remote and D2D IDM mechanisms for HRs Interoperability - V1, June 2019. https://www.interopehrate.eu/resources/#dels
[23] InteropEHRate Consortium, D4.2-Specification of remote and D2D protocol and APIs for HR exchange - V2, March 2020. https://www.interopehrate.eu/resources/#dels
[24] InteropEHRate Consortium, D2.5 - InteropEHRate Architecture - V2, March 2020. https://www.interopehrate.eu/resources/#dels
[25] “Android Studio”. Website: https://developer.android.com/studio
[26] “Dropbox”. Website: https://www.dropbox.com/h
[27] R. Saripalle, C. Runyan, and M. Russell, “Using Hl7 FHIR to achieve interoperability in patient health record”. Journal of biomedical informatics, vol.94, 2019.
[28] “ownCloud”. Website: https://owncloud.com
[29] “Google Drive”. Website: https://www.google.com/intl/en_in/drive/
[30] A. Kiourtis, S. Nifakos, A. Mavrogiorgou, and D. Kyriazis, “Aggregating the syntactic and semantic similarity of healthcare data towards their transformation to HL7 FHIR through ontology matching”. International Journal of Medical Informatics, vol.132, 2019.