N-Screen Provenance-Based Security Enforcement in mHealth

Richard K. Lomotey[*1], Ralph Deters[2]

Department of Computer Science, University of Saskatchewan, S7N 5C9, Saskatoon, Canada

[*1]richard.lomotey@usask.ca; [2]deters@cs.usask.ca

Abstract- **The use of mobile devices in the health domain (known as mHealth) to access the Electronic Health Record is gaining significant attention. Physicians are facilitated to access the medical data remotely and also run diagnosis in critical situations where the mobile usage is paramount. When the medical data or application is hosted on mobile devices (e.g., smartphones, tablets, etc.), it creates the need to provide protection against impostors. In the event that the mobile device ends up in the wrong hands, there can be undesired situations such as breach of privacy, impersonation, and the pollution of data. Moreover, physicians are facilitated today to use multiple mobile devices (n-screen) such as smartphones and tablets; and this requires serious attention on how user activity is tracked to ensure confidentiality. In this work, the methodology of provenance is explored to ensure that data access control in an mHealth environment is provided. Provenance is a methodology that maintains the life-cycle history of processes and data. The methodology is investigated and a variation of it that blends with policy-based access control in a mobile distributed environment is proposed. The preliminary results from testing the work showed high transparency in accessing the medical data, data protection, and security.**

Keywords- Provenance; Mobile Devices; Proxy; Rule-Based Access Control; Security; N-Screens

I. INTRODUCTION

The adoption of mobile technology across the enterprise landscape is gaining significant attention recently. Especially, smartphones and tablet devices can be employed as client consumer nodes to access data that is hosted on the Enterprise Information Systems (EIS) [1]. Today, most enterprises prefer mobile devices as data and information delivery nodes because these devices are portable and support user mobility; and those are two attributes that promote business productivity since remote access to information has become the order of the day in most enterprises. According to Gibbs' report, "The rise of tablets in the enterprise" [1], it is very clear that the future looks promising for the mobile commerce space considering the projected number of mobile devices in billions to be in circulation by 2015. The same report noted that most mobile devices in production nowadays support connectivity via varied network interfaces such as Wi-Fi, Bluetooth, and 4G and so on. While the mobile commerce space may be looking promising, other primary beneficiaries of the smartphone and tablet boom are the non-commercial enterprises such as m-health - where data is expected to be accessed over a secure wireless network for healthcare delivery [2].

In the medical sector, the use of mobile devices promises new opportunities for healthcare delivery. There is increasing number of patient-specific apps that aid the user to perform self-assessments, check eating habits, track physical activities, and so on. On the other hand, the beneficiaries of the mHealth paradigm are physicians and healthcare practitioners. They are able to access the Electronic Health Record (EHR) from their mobile devices and carry out diagnoses when required. This leads to advantages such as remote healthcare delivery, location-independent accessibility of medical data, and fostering strong relationship between patients and physicians [3]. Moreover, the consumer attitude today dictates that users own multiple devices such as smartphones, tablets, notebooks, etc. Furthermore, consumers want to have services and data accessibility across their devices. For instance, at the basic level, consumers want to access their emails across varying devices on the go as well as have services in synchronization across the devices. This requires the deployment of a new paradigm of applications, which is often referred to as *N-Screen* applications [4]. The deployment of N-Screen apps has also found its way into the mHealth field. For instance, medical apps that focus on facilitating data collection from patients outside the health facility should be deployed on cross platforms. In that case, when the physician on duty cannot use a particular mobile device due to unforeseen circumstances, another mobile device can be used for the same purpose. This then facilitates the physicians to have ubiquitous access to medical data regardless of their mobile device and location.

The problem that arises now is the enforcement of security in such n-screen enabled mHealth environment. The fact that the physicians can have multiple mobile devices that house medical records calls for the facilitation of data protection policies. If this is not enforced, there is the risk of losing information to unintended persons. This can lead to breach in privacy, medical data pollution, and attacks on the medical system.

In this work, the topic is investigated and the provenance methodology is proposed as a way to ensure medical services transparency in the mobile environment. A secure proxy layer that tracks the activities of the physicians and enforces some security policies is proposed. While provenance aids the tracking of the actions and activities of the physicians, the policy enforcement aids the enforcement of trust. The policy is enforced based on the combination of factors such as time, location, and the action that the user wants. For instance, physicians are required to provide further information if it is detected that they

DOI:10.5963/BER0302002

are in a drinking bar at 2:00 am and want to change a patient record on visits.

The remaining sections discuss some of the roles played by provenance in today's IT economy, the proposed methodology and framework, a preliminary evaluation of the proposed system, and conclusions.

II. PROVENANCE-BASED FRAMEWORKS

According to the works by Davidson and Freire [5] and Simmhan et al. [6], provenance implementation in scientific workflows can lead to reproducibility and services re-use since the methodology focuses on tracking services throughout its life-cycle. In enterprise information systems (EIS), data provenance can be implemented at the applications level, workflow coordination and control level, and system level [7]. The adoption of cloud services further fuel the debate on understanding the granularity of services security. By adopting provenance as a security measure, Zhang et al. [8] argue that cloud data and information leakage can be prevented through transparent and accountable event management. Hence, the authors focused on tracking the atomic operations on files in a cloud ecosystem such as creating a file, deleting a file, amending a file, reading from a file, moving a file, and renaming a file. However, the main contribution of the authors' work is the rule-based adoption for the provenance data traceability. The authors opined that data leakage can happen within the same domain (locally) or across domains. The local rule identifies activities such as file copying, file renaming, and file movements. The cross-domain rules however, track activities such as sender-side logic on the atomic operations, receiver-side logic on atomic operations, and email attachments.

Specific to relational database query aggregation, Amsterdamer et al. [9] proposed rules that govern atomic query operations. The authors argue that rules can be applied to operations such as single queries, multiple queries, and composite queries in relational databases. These rules can be defined to address the security issues and the complexities in query generation. Relational database provenance also proves to be efficient for ensuring the quality of the queried data. However, provenance can lead to storage complexity, time complexity, and storage size increment. Also, storing atomic provenance information can lead to data growth where the size of the provenance data can overtake the actual data size. These challenges have been the guiding principles for Bao et al. [10] who sought to reduce the challenges that provenance can introduce to relational storages. The authors noted that previous studies have proposed provenance tree techniques (as a means of tracing data derivations) but it can be time consuming to traverse the tree later. So, at first, the authors proposed rules that govern whether similar provenance data should be copied for identical tuples or referenced; and the authors favor the latter option. Then, the authors proposed dynamic programming methodology that optimizes the generation of a provenance tree. Additionally, the *Perm* (Provenance Extension of the Relational Model) framework developed by Glavic and Alonso [11] is meant to optimize the query pattern and storage space of provenance data in relational databases. For every piece of data generated, Perm automatically creates provenance data, which is also relational, so once a piece of data is queried, the provenance record of that data can equally be fetched without explicitly querying the provenance data for history derivation.

Furthermore, provenance is based on not only policies and rules, but also the roles that the individual system components play. She et al. [12] proposed a role-based provenance mechanism for services aggregation and trustworthiness. In distributed systems, actors such as hardware, software, and human operators are all considered as independent services. According to She et al. [12], provenance should be based on the roles that these independent actors play, especially when the actors are from different application domains. The composition of services from different application domains is challenging considering the fact that there is no centralized authority. Therefore, the approach proposed by the authors is distributed protocol access control. In this case, the authors considered the physical resources (e.g., files, directories, etc.), data resources (e.g., data and meta-data), and security authority that controls access. Furthermore, the authors explored inter-domain role-based access control (RBAC), which permits users in a domain to have access to resources in another domain through role mapping. In essence, the role that an actor is playing in domain *A* is translated to domain *B* when the actor moves to that domain. The data quality index in a RBAC is represented as a tuple of the format:

$$Quality_{Data} = (Reliability_{Data}, Trustability_{Data}, Reinforcement_{Data})$$

where $Reliability_{Data}$ is the reliability of the data from its source of origin, $Trustability_{Data}$ is the trustworthiness of the originators and contributors to the data, and $Reinforcement_{Data}$ is the reinforcement factor, which is based on the frequency with which individual system components repeat the generation of similar information.

There are also data provenance mechanisms that are based on group collaborations; they are an extension of the role-based provenance [13]. Enterprises keep local copies of provenance data but there are cases where data tracking is required between multiple enterprises. In that case, while local copies of provenance data can be kept at the individual participating enterprises, there is also the need to keep a global provenance record that aids in determining the activities of other users. In such group collaborations, it is important that enterprises keep sight of internal confidentiality of data so that privacy is not breached. The work of [13] proposes mechanisms that can aid in the local confidentiality maintenance as well as the global provenance tracking. The authors describe a uni-provenance mechanism where all entities have access to the environmental data and a multi-provenance mechanism where entities keep local copies of their provenance data and only make it available to collaborators from other domains on request and trustworthiness.

DOI:10.5963/BER0302002

Once provenance is enforced, it leads to the tracking of user activities, the recording of application procedures, and so on. In this regard, we can argue that provenance can potentially risk data and user privacy. Alharbi and Lin [14] proposed a privacy-preserving data provenance (PDP) mechanism that aids users to securely access services remotely without risking the exposure of their identities. In such environments, data provenance focuses more on the prevention of unauthorized user activities while relying on group signature to enforce user privacy. However, the PDP approach is feasible when there is a central authority (or trusted authority) that controls the security on the distributed trusted servers (the trusted servers in this case are the hosts and providers of services) and a huge user base. The same concern for securing provenance data has been the focus of Hasan et al. [15] who argue that provenance data can be forged or tampered with when it is passed through insecure networks. In this regard, the authors argue that provenance data should be encoded with cryptographic hash keys before they are transmitted.

While provenance services have been built to track data derivation and services as a user triggered event, there is also the need to look into automatic provenance mechanism. Braun et al. [16] proposed Provenance-Aware Storage System (PASS), which automatically collects and records provenance data without any intervention at the user and application levels. PASS records provenance data by observing the sequence of execution of processes (or events) at the operating system and kernel level in the Linux environment. Braun et al. [16] further noted that there exist Observed-Provenance Systems (where automatic provenance is enforced through observation of process execution sequence) and Disclosed-Provenance Systems (where users manually expound the workflow composition for provenance tractability). While the two types of provenance services are complimentary, the major challenge is that the identification of provenance granularity can appear differently at the process execution level from the user expectation. The question that arises then is how to synchronize provenance consistency between the user level and the system level. The authors then proposed versioning control techniques where changes at every level are recorded and a new version of provenance records is created once new updates are applied. The only critique of the PASS framework is that the conflict resolution in the versions is not discussed. However, the authors rely on provenance pruning to control the provenance record from growing out of proportion.

The question of what should be a provenance record and what should not is answered in the work of Reilly and Naughton [17]. The authors posit that provenance should be considered at two levels, logical provenance and infrastructure provenance. The former focuses on the transformation that is occurring to the actual data sets within the system while the latter focuses on the environment where the data transformation is happening (where the environment provenance can include operating system, date, processor, and so on). This notion is the underlying policy for the *Garm* [18] framework, which combines provenance between the data process and the system environment and the work presented by Lim et al. [19], who argue that data derivation history should be linked to the system environment.

Another mega trend in enterprise services deployment and consumption nowadays is collaborative services. The advancement in cloud technologies is facilitating the deployment of collaborative services that aid in workflow composition; however, the begging question is how tracking can be achieved in such environments [20]. This question is answered by providing a three-level provenance tracking, namely business level (using product lifecycle through product life status diagram), process level (using workflow instances through workflow action diagrams), and data level (using workflow instances through input, output, and conversion logic) [21]. Also, in a collaborative services environment, process documentation can be employed as a means to ensure trailing analysis [22]. The proposal of trailing analysis is to guarantee services tracking in a dynamic environment at runtime. The process documentation keeps the mapping relationship between the actors of the system, the messages that they are receiving, and the messages that they are sending. Furthermore, with the advancement in Web technologies, collaborative services that are Web-based can rely on provenance to determine the semantic structure of Web data [23]. Provenance record tracking in distributed services can be a daunting task due to the fact that the individual services and data are of different semantics, structures, and formats. To be able to deploy provenance technique for collaborative systems, a broker that converts the services heterogeneity into a common domain model can be implemented [24].

A. The Research Goal

From the extensive work on provenance, work is yet to be commenced on the adaptation of the idea to solve security issues on N-Screen application support. Primarily, we believe that any action taken by the physicians should be logged; and this is actually how most Health Information Systems (HIS) operate. Our focus therefore is on how secure accessibility of the medical data can be enforced in an N-Screen system, as well as the reduction of the risk of a request being made to the HIS by unwanted users due to an error or misplacement of the mobile device by a physician. Thus, we seek to explore the following questions further:

How do we detect unusual request from physician who own n-devices?

How do we know which devices belong to the physicians, and therefore should authorize medical data accessibility on those devices?

How should detected attackers be treated?

The answers to these questions are offered in conjunction with the Geriatrics Ward at the City Hospital in Saskatoon,

DOI:10.5963/BER0302002

Canada. We seek to answer the questions as an extension of the SOPHRA [25] project implementation.

III. THE PROPOSED PROVENANCE FRAMEWORK FOR N-SCREENS

The proposed system comprises of the mobile participants, a provenance proxy layer, and the main server farms that serve as the back-end infrastructure of the Health Information System (HIS). The healthcare professionals (HP) and the physicians can use multiple devices on which the SOPHRA application is running. The primary mode of communication between the back-end layer and the mobile nodes is over Wi-Fi and 3.5/4G. It is also important to state that the proxy and the HIS are hosted on a cloud computing platform. The entire architecture is illustrated in Fig. 1. The following discussions explain the composition of the various components.

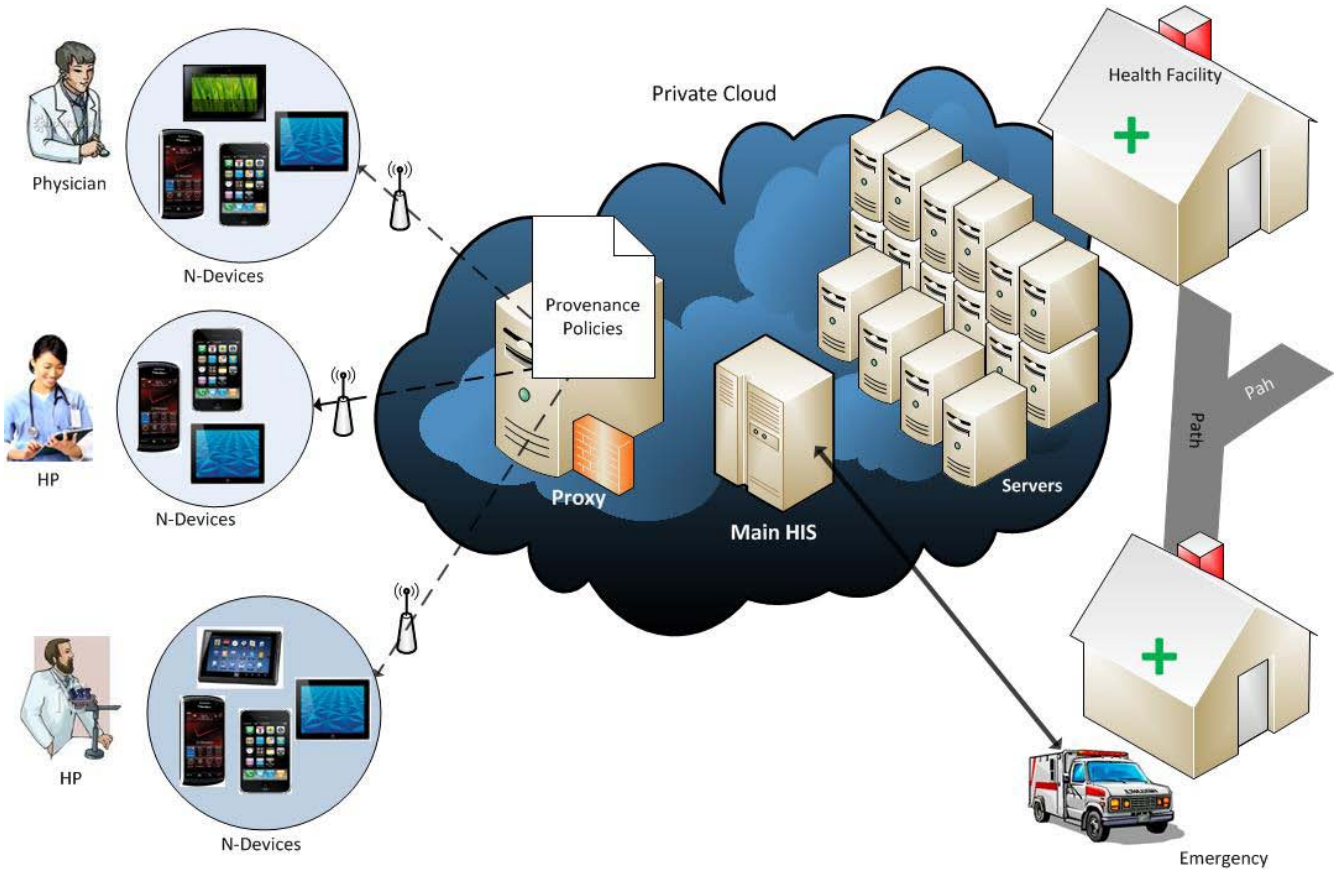

Fig. 1 The architectural design of the N-Screen mHealth environment

A. *The Health Information System (HIS)*

The Health Information System (HIS) is the main computer infrastructure that stores the medical data from the health facility (i.e. City Hospital at Saskatoon). It is where all the units within the hospital store their data for use. The HIS consists of several servers that perform different roles, such as:

- Database Server,
- Content Management Server, and
- Application Server.

Apart from accessing the services of the HIS from within the health facility, some of the units require remote accessibility due to the nature of their operations. For instance, the emergency unit works remotely from the health facility and constantly requires access to information.

In the present work, we shall interface our platform to the HIS. In the requirement to build the SOPHRA application, we seek to enable the healthcare practitioners to offer remote healthcare delivery. Especially, the Geriatrics patients can be supported remotely by the healthcare practitioners outside the health facility. The healthcare practitioners can go and visit the patients and retrieve records on the progress of how the patients are doing. In this regard, the SOPHRA application facilitates the healthcare practitioners to record information such as medications, visits, demographics, allergies, and so on. The same information can be modified by the healthcare practitioners as and when the need arises. This is the main reason that provenance and security is crucial for us, because unwanted users can have access to the device and modify the information. This is a threat to any mHealth system and however, the situation is further complicated with the advent of n-screen system

DOI:10.5963/BER0302002

designs.

B. The Mobile Participants

The mobile participants include the healthcare professionals (HP) and the physicians plus the n-devices that aid them in accessing the electronic health records. A mobile stores the medical data that includes information such as demographics, vitals, problems, medical history, and visits, as shown in Fig. 2. The healthcare professionals can access the records on the HIS, update the records as patients visit the hospital using the mobile application, and also add new records. In the implementation, two issues were investigated, how to design the cross-platform (i.e., n-screen) application and how to store the mobile data for later use.

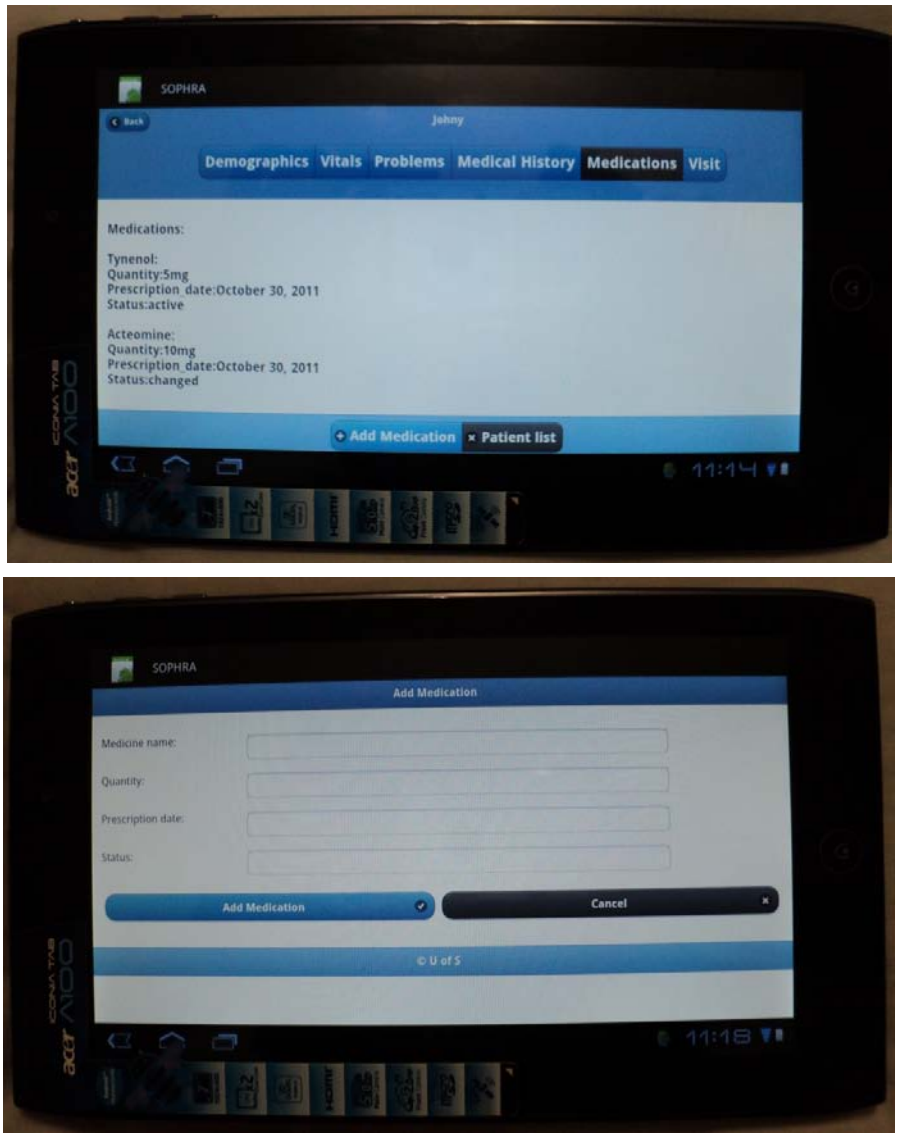

Fig. 2 Screenshots of the SOPHRA application

The design of the mobile application followed the mobile Web design, which enabled us to use web technologies for the development. The web is compatible with most modern mobile platforms, so it is convenient to build a mobile service following the web standard. However, mobile devices are not desktop browsers that have wide display. Hence, it is not advised to deploy web pages directly on the mobile. Furthermore, the web itself does not allow developers and researchers to have access to the native features of the mobile device such as storage, camera, etc. Therefore, the HTML5 was adopted to develop the application. Other web technologies such as the jQuerymobile, Cascading Style Sheets (CSS), and JavaScript were used to ensure a consistent look and feel of the application across different mobile devices. Thus, the application is a hybrid design that uses web technologies to render native application. It is important to state that the mobile application can also be designed following the native design approach. The native approach requires the development of the app in a platform specific language, e.g., Objective-C for iOS, Java for Android, C# for Windows Phones, etc. This means the same application has to be developed in different programming languages, which requires knowledge of all of these languages. This also can take significant programming time. Recently, there are tools such as Xamarin that allow the development of cross-platform native apps but the

DOI:10.5963/BER0302002

support is just at the business logic level, and not the user interface. This means the user interface has to be designed in language specific environments. The adopted approach however is efficient for the development of a single code base that can be deployed on multiple platforms such as iOS devices (iPhone, iPod touch, iPad), Android devices, Windows Phone, and BlackBerry.

Apart from the design environment, the mobile application follows the Model-View-Controller (MVC) approach. The view is the interactive layer (i.e. the screens as shown in Fig. 2) that allows physicians to interact with the application. The Model is the layer that stores the medical records on the mobile node to support offline accessibility and usage. Since mobile devices experience intermittent loss of connectivity, the mobile node medical record storage is proposed and it is accessible when the physicians cannot connect to the HIS.

C. The Proxy Layer

The proxy-layer was proposed to achieve most of the research goals on enforcing the access protection policy. As already posited, physicians can use multiple devices when accessing the medical records. This creates the need to investigate the best approach that can enforce protection of the medical data accessibility. The anatomy of the proposed proxy layer for the enforcement of provenance is shown in Fig. 3.

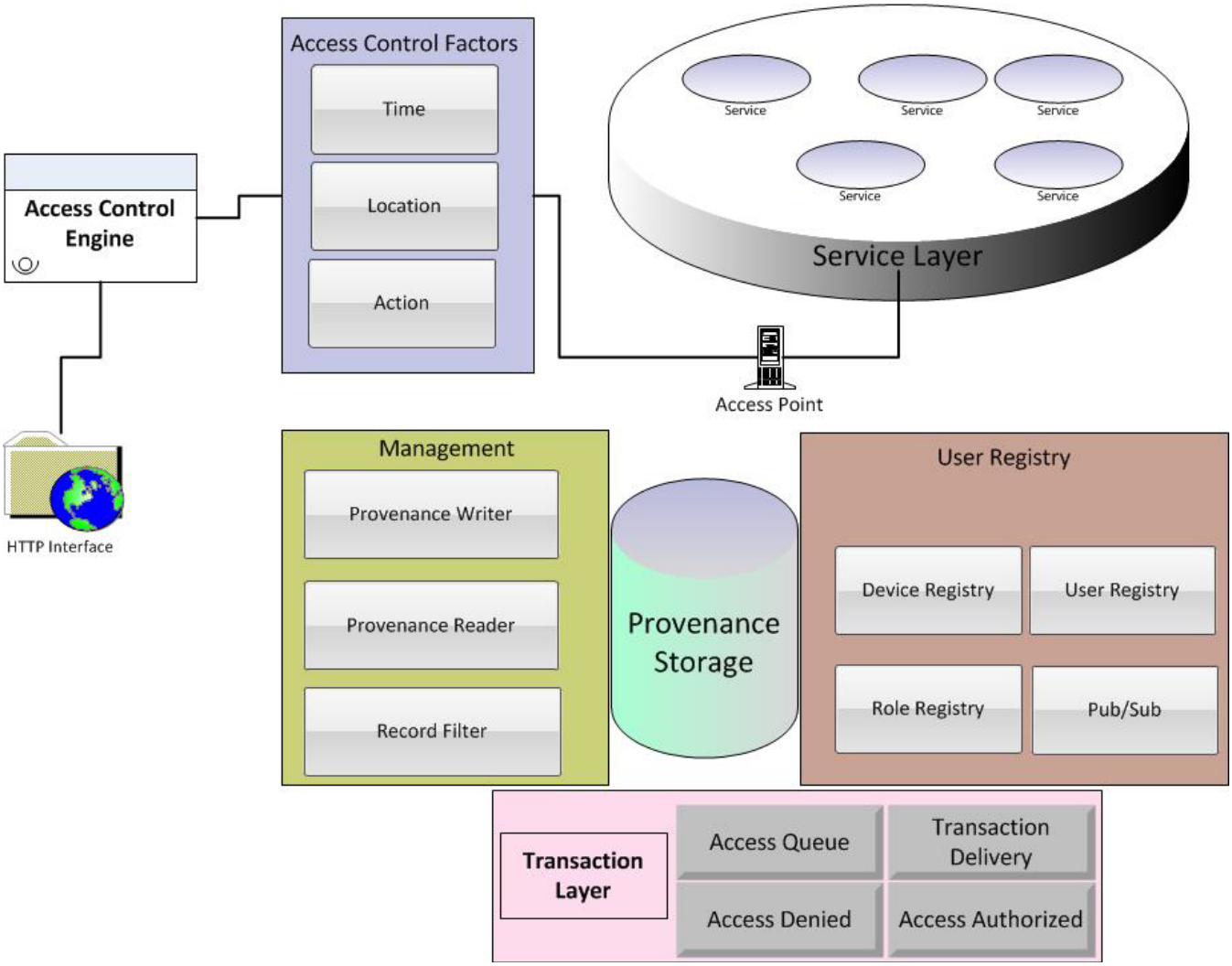

Fig. 3The anatomy of the provenance-based proxy

The main reason for adopting the provenance approach in the era of n-screen medical apps is to ensure services tracking. The provenance methodology as seen from the literature review facilitates the tracking of every action of users within a distributed system. In the proposed system, an HTTP Interface is defined and it allows the mobile participants to reach the proxy. The proxy acts as the message router that coordinates all the activities between the physicians and the HIS. Hence, the proxy has two HTTP interfaces that aid the mobile participants to send messages and the HIS to send responses. The explanations of some of the proposed components are as follows:

DOI:10.5963/BER0302002

A. Access Control Engine

This component is responsible for the determination of whether a request should be served or not. A request can be in the form of retrieving a medical data from the HIS, updating an existing record, or creating a new record. The system does not support the deletion request since the non-functional requirement of the application dictates that no medical record should be deleted. In the event that the system is hacked, attackers can send fake write requests such as create new records or update existing records, and this must be prevented. Hence, some policies were defined based on some specified factors in consultation with the project team.

B. Access Control Factors

There has been successful deployment of context-aware systems that defines system accessibility within a context. Context can be time-based, location-based, or role-based. In the present work, however, the policy is defined based on the combination of all three factors. Our access control policy is based on an algorithm that the proxy runs as shown below:

$$Access = Time \cap Location \cap Action$$

Even though the user is expected to provide username and password pair before logging into the system, this is not enough to enforce security since that requirement can be compromised. Thus, when a request is sent by users, the proxy determines the *time* of the request, the *current location* of the user, and what *action* the user wants to take. The combination of these factors is good enough for us to determine a genuine or suspicious request. To understand this further, let's consider the scenarios below:

1. A user sends a request at 1:00 am (i.e. time) from a drinking bar (i.e. location) to edit the allergy of a patient (i.e. action);
2. A user sends a request at any time from a different location other than Saskatoon to do anything;
3. A user sends a request at any time from any location to delete a medical record.

In scenario 1, the request is suspicious when the time and location of the user and the action to be taken are combined. Why will the allergy of a hospitalized patient be modified that late from a drinking bar? In this case, the proxy will present the user with a set of security questions that must be answered to justify the change. Furthermore, the proxy will not allow the modification to take effect on the HIS but the modification will be stored as a provenance record on the proxy until a supervisor/another colleague approves the changes. In the event that the request is sent by a hacker, the user will probably be able to answer the set of security questions. Better still, the changes will be discarded later when the physician and the supervisor reject the changes from the provenance record.

In scenario 2, the application designed is for use by the physicians at the Geriatrics Ward in Saskatoon Canada and their patients within a certain geographical boundary. It is also logical that the physicians can travel and carry their devices with them. The suspicious question however is, why will a request by a physician come from say Mexico? In this case, the request is stored as a provenance record and a communication is established between the physician and a colleague. The physician will have to justify to the colleague why the request is necessary; and this colleague can then approve the request. Until that is done, the request will be stored as a provenance request.

In scenario 3, the deletion request is not supported at all so this request will not be served.

There are other scenarios that are determined by the project team that takes these three factors into consideration and we keep updating the policy as and when they are defined.

C. The Service Layer

The service layer is the HIS where the medical records are stored. In reality, this component is outside the proxy layer. The proxy only accesses the services layer when a request is approved (i.e. passed the security access test). The proxy connects to the services layer through the *access point* which is an HTTPS interface. This ensures the communication is more secure.

D. Management:

Management layer determines how the provenance data is controlled. This includes the *provenance writer* – which controls all write operations on the provenance record, *provenance reader* – which controls how the provenance record is fetched, and the *record filter* – which aids in organizing the provenance data.

E. User Registry

The registry is the component that controls the details specific to the physicians. It is important to state that every activity of the users is stored as provenance record for the purpose of ensuring audit trail. In the registry, there are following sub-components:

- *Device Registry*: this is where the details of the n-devices of the healthcare practitioners are stored. The Universally Unique Identifier (UUID) of each device owned by the healthcare professionals is stored and mapped to the user. Thus, when a

DOI:10.5963/BER0302002

request is sent from a device with a UUID not associated with the user, the request will be declined. In this case, the healthcare professionals cannot just install the application on any unauthorized device.

- *User Details*: The user details include the physician's information such as names, username-password pairs, identifiers, and other bio-data that can facilitate the identification of the physicians.
- *Role Registry*: This is where the access roles of the system users are defined. For instance, some users can only view the medical records and as such, should not be allowed to make modifications.

F. Transaction Layer

This is where all the processes regarding the actions of the proxy are taken. There is the *access queue* which controls the support for incoming requests. These requests are served concurrently by the proxy. The access queue is then split into two, the *access denied* queue, where suspicious requests are stored, and the *access authorized* queue, where genuine requests are stored. The transaction layer queues the requests as they are delivered through the delivery *transaction queue*. However, the delivery of the requests follows a prioritization approach. The authorized transactions are handled and delivered before the access denied transactions are processed. In this case, genuine cases are treated with the urgency it deserves while suspicious cases are delayed.

In the next section, the proposed system is evaluated based on services provenance tracking and latency reduction.

IV. SYSTEM EVALUATION

The proposed system was evaluated to determine the performance of the proxy regarding the handling of request-response interactions. The system was evaluated using the following device: iPad 3 — OS: Apple iOS 5.1.0, Resolution: 2048x1536, Processor: A5X (dual-core, w/ quad-core graphics), Storage: 16GB, RAM: 1GB. The middleware is deployed on a privately owned cloud with the following specifications: Intel Core i-5 processor, CPU 2400@ 3.10 GHz 3.10 GHz, 16 GB RAM, and 64-bit operating system. The mobile devices connected to the middleware through 802.11g Wi-Fi 54Mbps connection.

The latency for processing read and write mix requests was evaluated. Assuming the physicians are sending a mix of requests that involve the creation of medical records and updates (i.e. write operations) and read requests, the time it takes to process the various requests was assessed. 1000 requests were issued and each request contained a mix of read and write operations. The requests might include 1000 reads only, 900 reads + 100 writes, 800 reads + 200 writes, ..., 0 read + 1000 writes. The result is graphed in Fig. 4 and the summary is tabulated in Table 1.

TABLE 1 THE WRITE AND READ MIX REQUESTS

Mean Inconsistent Window (ms)	Maximum Inconsistent Window (ms)	Minimum Inconsistent Window (ms)	Standard deviation	Variance
97.24	187.20	20.00	54.57	2978.03

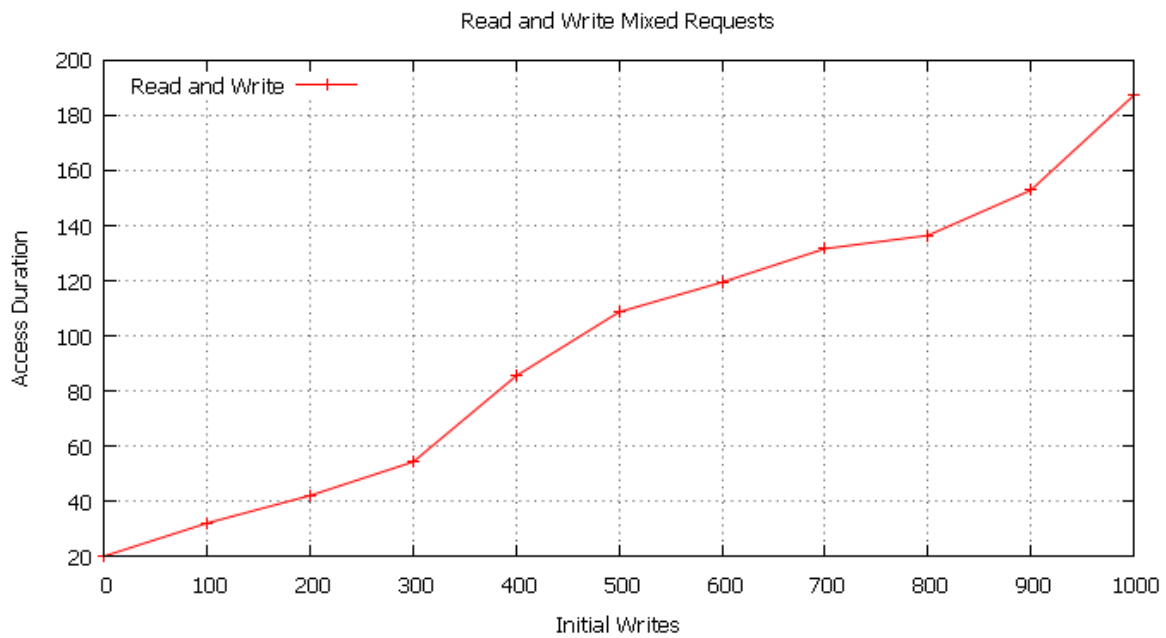

Fig. 4 The read and write mix requests

DOI:10.5963/BER0302002

The determination of the inconsistency window for the update synchronization is important. It was observed that the write operations took more time that the read operations to process. However, the processing of the write requests took a tolerable time within some milliseconds. The standard deviation was considered over a wide range of values from a minimum of 20 to a maximum of 187.2. This accounts for the high standard deviation at 54.57. The range further accounts for the high variance. Also, in the data set, most of the duration values were found to be closer to the higher values since we focused on bigger read/write operations. The observation however was that as the number of write operations increased, the time it takes to process increased.

V. CONCLUSIONS

Mobile devices usage in the health sector is on the rise with a lot of prospects being defined. The healthcare professionals are enabled to access the medical digital assets using their mobile. The collaboration of mobile technology and other ICT services is known as mHealth. Furthermore, physicians today use n-devices to access the electronic health record as a way of ensuring mobile ubiquity in the health domain. This however has led to the development of n-screen applications that can be deployed on cross-platforms for medical data accessibility.

While this may sound good, it also creates other problems such as threat to security. For instance, the mobile device can get into the wrong hands and undesired actions can take place. Moreover, the privacy of the medical data can be compromised in such situations when third parties are viewing personalized medical records. Thus, this work is in collaboration with the Geriatrics Ward at the City Hospital in Saskatoon, Canada, where the best approaches for the enforcement of security were explored.

In this work, the provenance methodology that enables us to enforce medical audit trail was proposed. We extended the SOPHRA [25] infrastructure where we provide a provenance proxy that keeps the states of a physician's n-devices, their account details, and so on. The proxy then relies on the combination of context information such as time of access, location of the user, and the required action to determine the integrity of the request. The approach has been deemed satisfactory by the project team. An evaluation of the proposed system further proves that the system processes requests within a tolerable time.

The future work will discuss the concepts of building a distributed proxy to answer more system level questions on scalability, fault-tolerance, and autonomic computing in the health domain.

ACKNOWLEDGMENT

We would like to thank the healthcare professionals at the Geriatrics Ward at the City Hospital in Saskatoon, Canada for spearheading and funding the project

Thanks also to the reviewers and the editorial team of Biomedical Engineering Research.

REFERENCES

[1] C. Gibbs, The Rise of Tablets in the Enterprise, GigaOM Pro, June 2011.

[2] J. Ranck, The Rise of Mobile Health Apps, October 2010.

[3] A. Prasad, R. Peterson, S. Mare, J. Sorber, K. Paul and D. Kotz, Provenance framework for mHealth, 2013 Fifth International Conference on Communication Systems and Networks (COMSNETS), pp.1,6, 7-10, Jan. 2013.

[4] R. K. Lomotey and R. Deters, Facilitating Multi-Device Usage in mHealth, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, vol. 4(2), pp. 77-96, 2013.

[5] S. B. Davidson and J. Freire, Provenance and scientific workflows: challenges and opportunities, In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (SIGMOD '08). ACM, New York, NY, USA, pp. 1345-1350, 2008. doi: 10.1145/1376616.1376772.

[6] Y. L. Simmhan, B. Plale, D. Gannon and S. Marru, Performance Evaluation of the Karma Provenance Framework for Scientific Workflows, Provenance and Annotation of Data, Lecture Notes in Computer Science, vol. 4145, pp. 222-236, 2006.

[7] O. Q. Zhang, M. Kirchberg, R. K. L. Ko, B. S. Lee, How to Track Your Data: The Case for Cloud Computing Provenance. 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), pp. 446-453, Nov. 29 2011-Dec. 1 2011, doi: 10.1109/CloudCom.2011.66

[8] O. Q. Zhang, R. K. L. Ko, M. Kirchberg, C. H. Suen, P. Jagadpramana and B. S. Lee, How to Track Your Data: Rule-Based Data Provenance Tracing Algorithms, 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1429-1437, 25-27 June 2012, doi: 10.1109/TrustCom.2012.175M.

[9] Y. Amsterdamer, D. Deutch, and V. Tannen, Provenance for aggregate queries. In Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS '11). ACM, New York, NY, USA, pp. 153-164, 2011. DOI=10.1145/1989284.1989302.

[10] Z. Bao, H. Köhler, L. Wang, X. Zhou and S. Sadiq, Efficient provenance storage for relational queries, In Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM '12), ACM, New York, NY, USA, pp. 1352-1361, 2012. doi: 10.1145/2396761.2398439

[11] B. Glavic and G. Alonso, Perm: Processing Provenance and Data on the Same Data Model through Query Rewriting. IEEE 25th

International Conference on Data Engineering, ICDE '09., pp.174-185, March 29 2009-April 2 2009, doi: 10.1109/ICDE.2009.15

[12] W. She, Y. I-Ling, F. Bastani, B. Tran and B. Thuraisingham, Role-based integrated access control and data provenance for SOA based net-centric systems, 2011 IEEE 6th International Symposium on Service Oriented System Engineering (SOSE), pp. 225-234, 12-14 Dec. 2011

[13] J. Park, D. Nguyen and R. Sandhu, On data provenance in group-centric secure collaboration, 2011 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), pp. 221-230, 15-18 Oct. 2011.

[14] K. Alharbi and X. Lin, PDP: A Privacy-Preserving Data Provenance Scheme, 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW), pp. 500-505, 18-21 June 2012.

[15] R. Hasan, R. Sion and M. Winslett, Preventing history forgery with secure provenance, Trans. Storage 5, 4, Article 12 (December 2009), 43 pages. doi: 10.1145/1629080.1629082

[16] U. Braun, S. Garfinkel, D. A. Holland, K. Muniswamy-Reddy and M. I. Seltzer, Issues in Automatic Provenance Collection, Provenance and Annotation of Data, Lecture Notes in Computer Science, vol. 4145, pp. 171-183, 2006.

[17] C. F. Reilly and J. F. Naughton, Exploring Provenance in a Distributed Job Execution System. Provenance and Annotation of Data, Lecture Notes in Computer Science, vol. 4145, pp. 237-245, 2006。

[18] B. Demsky, Cross-application data provenance and policy enforcement, ACM Trans. Inf. Syst. Secur. 14, 1, Article 6 (June 2011), 22 pages. doi: 10.1145/1952982.1952988

[19] C. Lim, S. Lu, A. Chebotko and F. Fotouhi, Prospective and Retrospective Provenance Collection in Scientific Workflow Environments, 2010 IEEE International Conference on Services Computing (SCC), pp. 449-456, 5-10 July 2010, doi: 10.1109/SCC.2010.18

[20] S. Sultana, M. Shehab and E. Bertino, Secure Provenance Transmission for Streaming Data, IEEE Transactions on Knowledge and Data Engineering, pp. 1, 2012. doi: 10.1109/TKDE.2012.31

[21] R. Genquan, Z. Li, W. Jianmin and L. Yinbo, One Method for Provenance Tracking of Product Lifecycle Data in Collaborative Service Environment, 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 347-356, 10-12 Oct. 2011.

[22] V. Tan, P. Groth, S. Miles, S. Jiang, S. Munroe, S. Tsasakou and L. Moreau, Security Issues in a SOA-Based Provenance System, Provenance and Annotation of Data, Lecture Notes in Computer Science, vol. 4145, pp. 203-211, 2006.

[23] U. Marjit, A. Sarkar and U. Biswas, A novel approach to develop linked data with provenance, , 2012 International Conference on Computing, Communication and Applications (ICCCA), pp. 1-5, 22-24 Feb. 2012, doi: 10.1109/ICCCA.2012.6179211

[24] M. A. Sakka and B. Defude, A mediator-based system for distributed semantic provenance management systems, In Proceedings of the 16th International Database Engineering & Applications Sysmposium (IDEAS '12). ACM, New York, NY, USA, pp. 193-198, 2012. doi: 10.1145/2351476.2351499

[25] R. K. Lomotey, S. Jamal and R. Deters, SOPHRA: A Mobile Web Services Hosting Infrastructure in mHealth, 2012 IEEE First International Conference on Mobile Services (MS), pp. 88-95, 24-29 June 2012, doi: 10.1109/MobServ.2012.14

Richard K. Lomotey is currently pursuing his PhD in Computer Science at the University of Saskatchewan, Canada, under the supervision of Dr. Ralph Deters where his main work focuses on mobile cloud computing. He has been actively researching topics relating to: ubiquitous cloud computing and the paradigm shift in enterprise mobility workforce support. Over a couple years, most of his works are industry collaboration with the Saskatoon Health Region.

Prof. Ralph Deters obtained his Ph.D. in Computer Science (1998) from the Federal Armed Forces University (Munich). He is currently a professor in the department of Computer Science at the University of Saskatchewan (Canada). His research focusses on mobile and cloud computing.

DOI:10.5963/BER0302002