Designing for Scalability and Trustworthiness in mHealth Systems

Sanjiva Prasad

Indian Institute of Technology Delhi, New Delhi, India

Abstract. Mobile Healthcare (mHealth) systems use mobile smartphones and portable sensor kits to provide improved and affordable healthcare solutions to underserved communities or to individuals with reduced mobility who need regular monitoring. The architectural constraints of such systems provide a variety of computing challenges: the distributed nature of the system; mobility of the persons and devices involved; asynchrony in communication; security, integrity and authenticity of the data collected; and a plethora of administrative domains and the legacy of installed electronic health/medical systems.

The volume of data collected can be very large; together with the data, there is a large amount of metadata as well. We argue that certain metadata are essential for interpreting the data and assessing their quality. There is great variety in the kinds of medical data and metadata, the methods by which they are collected and administrative constraints on where they may be stored, which suggest the need for flexible distributed data repositories. There also are concerns about the veracity of the data, as well as interesting questions about who owns the data and who may access them.

We argue that traditional notions of relational databases, and security techniques such as access control and encryption of communications are inadequate. Instead, end-to-end systematic (from sensor to cloud) information flow techniques need to be applied for integrity and secrecy. These need to be adapted to work with the volume and diversity of data collected, and in a federated collection of administrative domains where data from different domains are subject to different information flow policies.

Keywords: mHeath, smart phones, system design, scalability, privacy, metadata, contextual evidence, hyper-graphs, CAP theorem, eventual consistency, convergent replicated data types, distributed hash tables, trustworthiness, decentralised information flow control, structure-preserving hash functions.

1 Introduction

The term *mHealth* refers to the use of mobile phone technologies in the delivery of health care in a variety of settings [WHO11]. mHealth has become an attractive approach for providing better health care outcomes at lower cost and greater convenience to both the patients as well as to health care providers [GMS05].

R. Natarajan et al. (Eds.): ICDCIT 2015, LNCS 8956, pp. 114–133, 2015.

Such solutions have become viable due to significant technological developments, which include:

1. Cheaper, relatively reliable and more portable sensors of various kinds;
2. The wide adoption of increasingly inexpensive but computationally more powerful smart phones;
3. Almost ubiquitous cellphone and wireless coverage at low costs to consumers;
4. Cloud storage and computing technologies.

mHealth is of interest to both the developed world and the developing world since the same set of technologies involved can be adapted to work in quite different socio-economic environments. For instance, senior citizens or patients who need regular monitoring can be provided individual care at their residences by equipping them with a set of sensors, the readings of which are aggregated by a smart phone and transmitted at regular intervals or at times of emergency to a doctor, nurse or other caregiver in a remote hospital. Their condition can be monitored, reviewed and care advised or provided as and when required without their having to make periodic inconvenient and expensive visits to a hospital facility [WHO12]. (Such visits which take place in the current healthcare arrangements are often unnecessary and sometimes disrupt the equanimity and routine of such patients.)

At the other end of the spectrum, in developing countries with severe resource constraints, rural health workers equipped with a smart phone or tablet and a kit of sensors can periodically visit different settlements under their purview and take readings of various parameters such as height, weight, temperature, blood pressure, haemoglobin, pulse rates and even ECGs of say a couple of hundred villagers. They can also make *in situ* reports on the prevalent environmental conditions, documenting these with environment sensor readings and visual evidence taken with the camera on the phone. The health worker can report these public health data and medical data back to a primary health care centre, at which location this information can be analysed and eventually relayed back to a district hospital or public health research agency. While it may take some time for such a model to become reality [MBN$^+$10], kits such as the Swasthya Slate – an Android tablet and kit of medical sensors – have been built and deployed to provide effective low-cost health care to underserved communities [Swa12].

In the first setting, the purpose is primarily monitoring, with the sensors dedicated to collecting medical and environmental data of a single patient. Often the sensors are medical grade (and so compliant to the appropriate medical standards) and a high level of assurance about the quality of data is necessary. If the patient requires continuous monitoring of her condition, then constant real-time communication is required with the nearest treatment facility. Emergency situations also require immediate and real-time communication. The data collected need to be communicated and stored in the patient's electronic medical health record, with integrity and privacy [ABK12] being major concerns. In the rural health care setting, the focus is more on using the technology for screening and public health. One may probably not require extremely high fidelity and accuracy on the sensors if they are only being used for preliminary screening (thus providing an opportunity for low-cost innovations); however, they need to be rugged enough to work in challenging physical settings. In addition, the health care providers may not be highly trained or literate, and therefore the entire kit should be portable, easy to use and "fool-proof". If patients are not being continuously monitored, one may be able to accommodate some slack in the communication infrastructure, such as lower bandwidth or the ability to *tolerate delays.*

1.1 Scalability and Trustworthiness

In order to be widely adopted and effective, two critical issues that mHealth systems need to address are *scalability* and *trustworthiness*. Effective health care can be provided to a large populace only if the system conserves the valuable time of highly-skilled doctors at multi-speciality hospitals, by screening out those patients who with a high degree of assurance do not require urgent or immediate treatment. Regular screening and monitoring of community health can identify potential problems in individual patients (as well as communities) well before their condition develops to a point where treatment is both more expensive and possibly less efficacious. Low-cost medical sensors reduce the cost of deployment though usually by sacrificing accuracy of the medical readings. However, if these were to be used only for screening, then they can potentially improve healthcare. Coupled with a reliable communication system that links the patient readings taken in the field to data bases of electronic medical records, the entire model of health care delivery may be radically improved. Scalability means that the system can deal with a huge *volume* of data collected over a wide geographical area over a sustained period of time. The data collected may display a wide *variety* in the kinds of medical parameters monitored, as well as in formats in which they are presented. Finally there is a need to *validate* the data being collected, and to ascertain their *veracity.* Otherwise, we would be saddled with large amounts of data of dubious quality, on the basis of which no sensible or effective medical decisions may be possible [OKOG12].

This paper presents some of our early learning experiences in trying to develop parts of an mHealth system such as developing low-cost sensors, collecting information from the sensors in a systematic manner, and transmitting it using technologies available on stock mobile phone to standard repositories currently being used in hospitals and regional health centres. As stated above, our primary interest has been to ensure that the solution is scalable, and that there is a modicum of quality assurance in the data that have been collected. We do not present here any new technical results in the theory of distributed computing, but suggest that certain ideas that have been proposed appear to be promising in engineering a workable solution for mHealth, under a collection of "legacy" constraints.

The rest of this paper is structured as follows. In the remainder of this introduction, we discuss some of the requirements of an mHealth system that we envisage, focussing only on some aspects related to scalability and trustworthiness. We do not address here several other aspects of the system which are addressed using standard techniques well established in the distributed computing literature. In particular, we do not delve into network design and communication protocols. Nor do we discuss a variety of distributed data base issues such as fault-tolerance and efficient information retrieval. In the domain of security, we do not concern ourselves here with the algorithms used for encryption and hashing, or about access control mechanisms. We believe that many of these issues have been addressed by existing software systems; moreover, one of the design constraints under which we operate is that it would be too ambitious and also infeasible to redesign the entire healthcare information system. Instead we concentrate on what security- and storage-related ideas will allow the design of a decentralised, interoperating federated collection of extant systems that have been deployed, with minimal modifications and as modest a trusted computing base as necessary to achieve a reasonable (but by no means absolute) degree of trustworthiness. We indicate how the architecture of the system must avoid common fallacies encountered in the design of distributed computing systems. We do not attempt in this paper to survey the field of mHealth systems since there already are good surveys of the area [WHO11].

In §2, we present the notion of a *medical encounter* as the basic unit of gathering medical information, and indicate why it is important to include metadata concerning the context in which those data were collected. We present a few examples of how such metadata may be used to answer questions in the medical domain, as well as in domains related to the administration of health care and to research in public health. Following that, in §3 we address the question of how data and metadata should be stored, suggesting hyper-graphs as a suitable model for representing the data and their interrelationships. We mention how data repositories may be distributed across different places and administrative domains and possibly be of very different character. Furthermore, different fragments of a single data record may be spread across these distributed repositories. We indicate that graph-oriented distributed hash tables offer an efficient solution for accessing values from such distributed data repositories. We end the section by suggesting that *eventual consistency* provides reasonable semantics, and that *conflict-free convergent data types* can be implemented in mHealth settings without incurring prohibitive overheads in achieving this weaker degree of consistency between replicated copies of distributed fragments of a medical record.

In §4, we address the other major issue on which we have focussed, namely privacy and integrity of medical data (and metadata) [ABK12]. We argue that merely securing communication using encryption and storage using access control is inadequate when different principals exchange information sensitive to them, especially across different administrative domains. We identify the possibilities of security being compromised due to information flow between different applications in different components of the system, and examine techniques from information flow control [Den76] that have been extended to decentralised settings [ML98, KYB^{+}07] for end-to-end information flow control. We mention how mechanisms for information flow control can be systematically incorporated into the “stack” developed for collecting data and contextual metadata, and mention some of the issues in building suitable security infrastructure into the mHealth system. We conclude in §5.

1.2 Requirements

We identify some common requirements which should be satisfied by a large-scale distributed mHealth system involving people, devices and communication and storage infrastructure:

- **Sensors:** These need to be robust, efficient, reliable and easy to use by lay persons (e.g., the patient or her family, a not-very-literate health worker). There already are several low-cost sensors that can be used for screening. However, these “stock” sensors often have no built-in communication facilities (Bluetooth LE), and may not be of medical grade; nor do they have any security features.
- **Configuration:** Configuration for collection of the data should be simple and not have many device dependencies. The sensor kits should easily connect with a smart phone, exploiting the computing power there for all the necessary analysis of the readings, collation of related readings, etc. Upstream communication can then be from the smart phone using a variety of options (3G, 4G, WiFi,...), and preferably opportunistically, using the most appropriate medium based on cost, time-criticality, importance and availability.
- **Communication:** The choice of communication media and protocols should not be hard-wired into the solution. The system should be neutral about the particular media used, especially since such a system is expected to operate for several years, if not decades, in the face of ever-changing technologies.
- **Data Representation Formats:** The data (and metadata) collected display great variety. Readings can be
  - discrete symbolic readings such as “clear”, “bloody”, “murky”, etc.;
  - discrete numerical values such as temperature or blood pressure;
  - sampled readings taken over an interval;
  - waveforms such as ECGs;
  - graphical images such as X-rays;
  - audio or video recordings, MRIs, etc.

  A versatile system should be able to represent all these various types of data, from the collection phase to their storage in a data repository. Moreover, since these data are intended to be long-lived (the lifetime of patients, if not longer), the encodings for interpreting their bit representation should perhaps be encapsulated within the representational formats. Encryption for privacy, and hashing for anonymity add further complexity to the endurability of any data representation solution.
- **Interoperability and Seamless Integration with Medical Records Systems:** Medical data collected must be converted into a standard electronic medical record (EMR), health record (EHR) or patient health record

(PHR). Since different hospitals may have invested in data bases and hospital information systems (whether proprietary or open source), and are unlikely to convert to yet another system, the design of the system must be agnostic with respect to the data repositories.

- **Security:** If an mHealth solution is to be acceptable to the public, it must provide a degree of data security expected by patients and users of the system. Moreover, health care providers require a high degree of integrity and trustworthiness of the data collected using mobile sensors, since the collection of health data is removed in time and space from the usual hospital setting where doctors may trust their staff, equipment and facilities (laboratories, information systems) to comply with standard procedures.
- **Interactive Queries:** The data collected using such a widespread and diverse system must support interactive querying. This also requires appropriate decentralised organisation of data respecting administrative boundaries and ownership policies and access control, with suitable semantics regarding consistency of data.
- **Fault Tolerance:** The system is expected to work in an operating environment where there can be a variety of faults. An appropriate adversary model needs to be defined, that can capture the various kinds of failures with respect to with the system is resilient.
- **Legacy issues and compatibility with existing systems:** In addition to network and data bases, as well as security policies, there may be a host of other legacy issues which any mHealth solution must respect.

1.3 Avoiding Common Fallacies in System Design

The architecture of the mHealth system should perforce avoid common fallacies about distributed systems:

- *We do not assume that network at any layer is reliable.* In particular, our experience with protocols such as Bluetooth used in collecting sensor readings is that users often do not configure the connections securely, and that connections may break due to mobility of devices (sensors, phones), or electromagnetic interference. We also note that sensors may slip from their ideal position while taking a reading, or that a particular protocol for taking readings may not have been properly followed (e.g., ensuring that the patient should be seated, at rest and readings are not taken immediately after vigorous activity, and that the cuff is at heart level when taking a blood pressure reading). The challenge there is met by designing a network protocol stack that sets up a secure, reliable connection between the sensor kit and the Android device over protocols such as Bluetooth, with different layers of the stack dealing with elicitation, validation, provenance or contextual information and security [KGPP14].
- *We do not assume that communication latencies are zero or even negligible. Nor do we assume that there is unlimited bandwidth, or that transport costs are zero.* In fact, the system is based on the premise that it must work

properly in the face of being often disconnected, and most portions of the system are configured to be delay-tolerant, when permitted by the application. The communication protocols are designed to avoid wasted bandwidth and dropped packets.

- *The network is not assumed to be secure.* Security is an important concern in any healthcare system, particularly the privacy of sensitive information, and more importantly the integrity of the data (and metadata) collected. A major part of our ongoing research lies in defining appropriate models of security by identifying attacker models and mechanisms that can ensure end-to-end secure flow of information permitted by reasonable sets of policies. We do not assume that there can be a centralized solution to security when a variety of different principals are involved.
- *The topology of the system is not assumed to be static.* In fact, mobility – of sensors, devices and principals – is a defining characteristic of an mHealth system. Accordingly, we do not try to embed rigid routing policies into the communication protocols and structures. Mobility also has important consequences on scalability, especially in dealing with namespaces and routing tables, as well as with security and trustworthiness.
- The most important realisation we reached in the design of a healthcare system is that the system is decentralised not merely in space but also in that *there is no single administrator either of namespaces or of security policies. There is great diversity and autonomy* in the different health care organisations (hospitals, research organisations, etc.), each with their own data security policies, different data base access control mechanisms and information disclosure and privacy policies.
- *The network is not assumed to have a uniform or homogeneous structure.* The communication media, bandwidths, protocols, etc. exhibit great variety. Clearly at the peripheries, especially in the developing world, the network is slow, "flaky" and often inaccessible; communication may be over mobile phone carriers. On the other hand, within a hospital or a research organisation it may be over reliable wired high-speed optical fibre networks.
- We realise that it will be prohibitively expensive as well as infeasible to attempt to design a uniform solution that can be adopted by all principals and organisations involved, so the focus has to be on developing a system that *works with the pre-existing infrastructural arrangements* chosen by the different healthcare organisations involved (network, operating systems, data bases,...), supplementing them with components that can ensure a degree of trustworthiness and value-addition while ensuring scalability.

2 Healthcare Encounters

Consider a developing world scenario where health worker Heena, equipped with a kit of sensors and a smart phone, visits a patient Puja in a village, and takes her readings for body temperature, pulse, blood pressure, weight, haemoglobin, and an ECG. All of these readings are annotated with Puja's ID, Heena's employee code, the time and place where the readings were taken. In addition, a few environmental parameters such as the ambient temperature, pollution levels, humidity, etc. may be captured by sensors. Also consider a personal healthcare scenario, where a set of sensors worn by the patient constantly collect and send to a medical facility or doctor via a smart phone readings of the patient's heart rate, temperature etc., together with readings from an accelerometer worn by the patient. The capturing of all this information taken together constitutes a *healthcare encounter*, the outcome of which is a single record consisting of the various medical data readings bundled together with critical metadata (who, whose, when, where, with what, etc.) regarding the context in which the readings were taken. An encounter is the basic unit around which the mHealth system records may be built. A patient's medical history may then be viewed as a collection of records produced by such encounters.

However, a more useful metaphor is that of a "conversation", where the encounter records are utterances but following which connections and correlations may be made between this encounter record and those of, say, the patient's earlier encounters, or of encounters of the patient's family members, or others in the same locality, or those taken by the same health worker Heena, etc. In other words, if the encounter results in a record in a data base, various "meta" records are created by various analyses that are performed either on the smart phone or in the hospital or at a regional level in a public health researcher's data base. Such meta-level observations are similar to commentary about previous utterances, or "asides" between a subset of listeners that may or may not be accessible to all the participants of a conversation. By making such meta-records first-class in the data repositories, we obtain a rich information system which may be queried in different ways for greater effectiveness.

The richness of this system may be successfully exploited by representing the data as a federated decentralised data repository. There have been various proposal's (e.g., by the NHS of the UK) to build centralised data repositories of all patients, which can be accessed by various health care providers. However, these proposals have assumed that the system would be highly centralized, and operating within a single administrative domain. While there are obvious benefits (to the patients, to the nation, to caregivers, to insurance companies, ...) of building such a system that exceed the costs involved, they have met with resistance from both caregivers, who may possibly be wary of the disruptions involved in migrating from their existing systems, as well as privacy advocate groups who worry about the compromising and exploitation of sensitive personal information of patients. Moreover the complexity of designing such a system and having it adopted uniformly is perhaps its greatest drawback.

2.1 Metadata Matters!

The metadata collected by various sensors including those on a smart phone play an extremely important role in effective decision making, both medical and administrative. An ultrasound image is of little value to a doctor unless she know whose it is and when and where it was taken. There are also other uses of the metadata, some of which may include:

- If Heena is paid by the number of patients she visits, she may have an incentive for defrauding the system by uploading readings from a small set of locations passing them off as readings from far-flung villages. Or she may be passing off old readings as new ones. If the metadata associated with an encounter were bundled in a secure, nontamperable manner with the readings, such fraud may easily be detected.
- The readings coming from a particular device may be observed to be consistently lower that an expected range. This may be for a variety of reasons: the health worker or patient has not placed the sensors properly; or there is a fault in that particular device; or a design and/or manufacturing problem with a particular batch of devices of particular brand; etc. Having metadata related to the context of the encounter can help detect problems with the data collected, and appropriate corrective actions can then be initiated;
- Researchers may be able to query anonymised data, correlating them across time or space or other demographic information. This requires being able to perform statistical analysis “underneath” the anonymising transformations on the data.
- A device manufacturer may be able to keep track of the deployment, use and performance of various devices that they have manufactured and sold to the health care providers.

Many of these metadata are implicit in traditional hospital-based care, where the trustworthiness of the data derives from it being collected in controlled contexts that provide the medical practitioner or administrator a high degree of quality assurance. This assurance of quality is what is lost when the encounter is removed in time and space from the consumer of the information. An important design consideration is therefore to record sufficient *contextual evidence* for the data to enable informed decision-making [PPM$^+$13].

2.2 Data Collection

Most sensors do not come equipped with facilities to collect the contextual evidence metadata. We have proposed and prototyped systems [KGPP14, KG14] where a set of sensors are connected via a micro controller board (Arduino currently, but Raspberry Pi or Intel’s Galileo may be used) to take readings from the sensors in a coordinated fashion and communicate these to an Android smart phone over Bluetooth, Wifi or even physically via USB. An Android app running on the smart phone initiates the process of taking readings during the encounter, and establishes connection with the micro controller board, which then elicits readings from the sensors, and bundles them into a communication packet that is sent to the smart phone. The readings are examined for validity and some corrective actions are initiated locally. Various parameters are set by the Android app using the phone sensors (e.g., GPS, time, the camera parameters).

The protocol has been organised as a "Sensor Stack" with layers for connection establishment and management, elicitation, validation, and adding contextual metadata. Security layers for secure information flow are currently being incorporated. The Android app finally compiles the data and metadata collected during an encounter into a commonly interpretable format (XML in our case) suitable for communication to a variety of EMR/EHR systems.

3 Data Base Design Issues

Due to the great variety in the kind of information being collected, a uniform template-based representation is probably *not* the best approach to representing medical data records. New kinds of information are likely to be added. Of particular interest to us are the metadata that constitute "contextual evidence". These may not be of interest to the hospital (and thus no provision may be made for such information in their health care records). The data generated in an encounter seem to be semi-structured. There also are temporal and causal links between records. Yet various electronic medical records systems are built over relational data bases such as SQL and MySQL. As noted earlier, it is unlikely that a hospital will migrate to a new kind of data base even if the benefits are apparent. Moreover, the system needs to accommodate several hospitals, each of which may have their own preferred EMR system. Therefore, one needs a flexible, EMR-agnostic way of incorporating the extra (meta) data into existing installed data bases. A solution is to build a separate data repository for the metadata and then placing references to and from it in the various hospital EMR systems. This has the advantages of:

- permitting more flexibility in the representation of contextual metadata, which may even be represented as key-value stores;
- decentralised management of the information, with each hospital as well as even patients controlling certain information according to their policies, and managing these policies locally;
- flexibility in expansion of the system, incorporating new kinds of data bases, more hospitals and a variety of healthcare/medical record formats;
- more efficient query-processing by exploiting locality of storage.

However, to realise such these advantages, one needs a model that is general enough to capture and anticipate the various usages. We have already alluded to the kinds of correlations between different medical records that a doctor, patient, researcher or administrator may make. Such questions translate into queries on data and metadata, and even on information derived from analyses done on them. It is our position that *hyper-graphs* provide the correct abstraction for the representing the information. Hyper-graphs are graphs where an edge can connect more than one node. Moreover, they can easily accommodate higher-order hyper-edges, that connect hyper-edges. Currently there are a few hyper-graph data bases, e.g., [Ior10]. Some of the advantages of hyper-graph-based data bases are that they:

- provide a powerful medium for data modelling and knowledge representation;
- can express n-ary and higher order relationships between graph nodes;
- provide for graph-oriented storage, and so can support graph traversals and path-queries as well as relational-style queries;
- can support customisable indexing and storage management; and
- can accommodate extensible, dynamic data base schema through appropriate typing.

The typing frameworks and query language design issues here provide interesting problems for programming language researchers.

3.1 Distributed Data Base Issues

Apart from representational issues, this being a heterogeneous distributed system, with a high degree of mobility, asynchrony, concurrent operation and possibility of a variety of failures, there are significant problems related to concurrency and fault-tolerance that need to be addressed in the design. Any centralised solution is not an option. The system comprises numerous smart phones connected to one another and to health care centres and their data repositories "in the cloud", all working asynchronously; so any approaches based on synchronisation are precluded.

The first issue is that of atomicity, which immediately raises the question about what constitute transactions in such a system. A candidate answer is "the collection of information during an encounter". *It is reasonable to view an encounter as the basic unit for updates.* However, what is the span of an encounter? Does it complete with the collection of readings by the micro controller board, or when the information is transferred to the Android app, which makes an XML (or similar) record for upstream transmission? Or does it conclude when the information is uploaded to the cloud or to a hospital repository? We have already noted that the data and metadata within even one record may be distributed spatially (due to ownership and administrative constraints). Thus it is not entirely trivial how such a distributed write should be atomically executed across the different sites involved. Thus the atomicity of transactions is not an easy question, and the possible answers have an immediate bearing on notions such as consistency.

Indeed we confront the issues of consistency, availability and operation under network partition, as in Brewer's so-called CAP theorem [Bre01]. Network partitioning should be assumed to be the normal mode of operation, and availability of the system is essential for its functioning. Therefore, we have to relax the requirements of strict consistency. Indeed, it seems that working with a notion of *eventual consistency* yields workable solutions which do not require a significant overhead, and support autonomous functioning of entities in the mHealth system. Fortunately, in the case of medical records, we can exploit the fact that *no records should be deleted.* Readings that have become irrelevant, e.g., a normal body temperature reading from last year, or even readings known to be erroneous are preserved in the store; they may be *deprecated* and *discarded* during medical decision making by a doctor, but are still maintained for historical reasons, or to answer meta-queries unrelated to the patient's health care (e.g., did the thermometer work correctly, or was the reading taken as prescribed in a healthcare protocol?). Thus we can operate under the assumption that information in the system grows monotonically. (In practice, there may be a hierarchy of storage, with old, irrelevant, wrong and other such data that are unlikely to be useful banished to lower, slower rungs of the hierarchy).

Recall that the contents of a logical record may be distributed across many repositories. In distributed records, some information needs to be replicated across different repositories, both for efficiency, and to be able to link information in different parts of a hyper-edge which are stored on separate repositories. This replicated and cross-referencing information must necessarily be maintained in a consistent manner. Writing of semantically unrelated parts of a single record into distinct repositories allows a degree of flexibility (e.g., commutation) in the order in which writes may happen in a single distributed write transaction.

The monotonicity assumption makes it possible to design the system to exhibit local coherence and eventual consistency. Causal linkages must of course be preserved (these are recorded perhaps as "why-provenance" information [BKT01] in some hyper-edge) but not all temporal precedences are relevant for justifying treatment decisions made on information available at a particular time and place. Thus if we can ensure that updates to hyper-edges can commute, and all of its replicas execute all updates in causal order, then the replicas can (eventually) converge without invoking any elaborate coordination mechanisms. Monotonicity of information and adopting weaker notions of consistency thus make it possible to organise the health records as a (bunch of) Commutative Replicated Data Types (CRDTs) [SPBZ11]. CRDTs have the excellent property that they eventually converge without any complex concurrency control mechanism. They are ideally suited for extremely large information systems; the principal difficulty in designing CRDTs lies in efficient representation of the data type to have good local coherence and convergence properties across replicas, with respect to the series of operations performed on them. Write-monotonicity and read-only operations in the hyper-graph data type greatly facilitate building a large, scalable distributed data repository with good fault-tolerant properties. We have previously experimented with a prototype implementation of hypergraphs as a CRDT [Pri11], with promising results, and intend to incorporate these ideas into our prototype mHealth data repository implementation.

3.2 Accessing Records

Even with migration of data to the cloud, there is a need to have fast and reliable protocols for accessing records. We assume that, in general, the minimal organisation of data in any repository will be some variant of a *(key, value)*-store; any additional structure will be built on this basis. As we assumed that the basic records in the system will be encounters, each encounter will be given a unique key, perhaps derived from the personal IDs of the participants and the time and place of the encounter. Using suitable hashing functions, a key may be derived, which can be used for locating the record using Distributed Hash

Tables. We envisage using a DHT based on ideas from systems such as Chord [SMK$^+$01] or Koorde [KK03] which have a high degree of scalability and which exploit graph-theoretic properties to make retrieval more efficient and robust. In particular, given that many queries will involve some commonality of information between the records sought to be retrieved (the same patient, or the same health worker, or spatial location, or time period), we plan to use *locality preserving hashing* techniques, which map closely related records to the same or nearby repositories, thus making retrieval more efficient.

3.3 Presentation of Information

The more interesting challenges lie in how information is to be presented to various consumers of information. It must, of course, conform to the access privileges and privacy policy pertinent to that information. Moreover, the consumer of the information (doctor, administrator, patient, ...) must not get overwhelmed by the entire record, with data and metadata, and should also not be presented with a complete medical history of a patient *as recorded, replete with loads of dated, irrelevant, deprecated information.* This comprises a whole set of interesting research problems in data science, information retrieval, security and query processing.

A further issue that needs greater conceptualisation concerns the long-term preservation of the data. The interpretation of bits (that represent records) lies embedded in the software used to create and read the records. One has to design for preservation of backward compatibility whenever the software is upgraded or when any component of the system is upgraded or replaced. Alternatively, all data should come with a generalised self-strapping protocol using which the interpretation of bits is never mutilated.

4 Ensuring Trustworthiness

In any large distributed system dealing with such voluminous data, it is necessary to ensure data integrity (that it not be tampered with) and also that the sensitive information of each principal in the system is not divulged to any unauthorised party. Data stored in the data repositories (data bases, file systems, key-value stores, etc.) are said to be "at rest". Such data are secured by using access control mechanisms, which are supported by the operating systems and/or data bases. When data are"in motion", i.e., when that information is being communicated between systems, security protocols based on modern encryption techniques are used so that an attacker (active or passive) cannot compromise the integrity of the data, or learn secrets etc. However, just these two sets of techniques are inadequate to ensure that sensitive information is not improperly divulged, nor that information that is trusted is derived from untrusted information sources and untrusted data. The leakage, as they say, happens "at the joints", namely from the applications that access the data stores, process the information, and then put the results onto communication channels, transferring information from one administrative and security domain to another. Very often the information is leaked implicitly (i.e., the secret is not explicitly divulged, but can be inferred by the adversary from information accessed by it that is derived from the secret information).

It is our contention that ensuring trustworthiness of an mHealth system requires addressing security not just of data at rest and data in motion but also of data during computations. In other words, ensuring secrecy and privacy are end-to-end design issues [SRC84]. Security cannot be ensured piecemeal even if one were to use the best techniques and implementations for individual components; the users of the mHealth system should be able to specify and rely on the system to correctly deal with privacy and integrity of their data.

4.1 Information Flow Control

The problem of programs leaking information or computing results from untrusted sources is addressed by the techniques of *Information Flow Control* (IFC) [Den76] . Programs are analysed to check whether during their execution information can flow from data sources (input variables, files, etc.) considered secret to public or insecure sinks (output variables, output files,...). Dually, for integrity, the analyses check whether output values that are trusted are dependent from data that are considered untrusted. The analyses can be at run-time (dynamic), prohibiting accidental disclosure for instance. Alternatively, the analyses can be performed statically (at compile time) and programs certified as secure or (conservatively) labelled as insecure [DD77]. The analyses assume that security classes form a lattice, and permitted information flows are those conforming to the lattice structure. Each programming language primitive is abstractly interpreted in terms of meet and join operations over this lattice.

IFC analysis at the programming language level is a fine-grained analysis technique to ensure security (whether privacy or integrity). It assumes that the source code of the entire program is available, and that the entire program executes within one security administration domain. For large-scale distributed systems with thousands of principals each with their own labelling of particular pieces of information as private, such an analysis is not viable. IFC techniques were modified by Myers and Liskov to work in a *decentralised* label management framework to protect data for different users, each with their individual policy [ML97]. These DIFC techniques, which work by each principal labelling its information (without there being any single security authority) [ML98], have been implemented at the operating system level in systems such as Asbestos [EKV$^+$05], HiStar [ZBWKM06], Aeolus [CPS$^+$12], and Flume [KYB$^+$07], the last of which introduces the concept of an *endpoint*. DIFC techniques have also been adapted for data bases [SL13].

4.2 Security Model

Before we describe how DIFC techniques need to be further adapted for mHealth applications, we briefly describe the security model. Characterising the *adversary* is one way of understanding the operating environment in which the system needs to be able to function. (The notion of an *adversary* is not merely one that we encounter in security literature; it is perhaps a fundamental idea for understanding the limitations of a computational system, whether in complexity theory or in failure models, etc.)

The environment includes typical adversarial behaviour for any communication protocol (not merely a cryptographic one). Messages between components of the system can get lost, duplicated, corrupted and an eavesdropper may attempt to analyse (by decryption using available keys, or even brute force attempts to learn the inputs from the output of a function) a message to learn its component contents. The adversary can also fabricate messages using any keys, nonces, hash functions and any other available data. We may consider a variety of adversarial behaviours, from the Dolev-Yao model that assumes perfect encryption to computationally-bounded and resource-bounded adversaries. We have not considered *denial-of-service attacks* though these are a real possibility in mHealth systems. (At the very least, we should ensure that components of the system do not flood others with unboundedly many messages.)

We require that the system function properly even if some components fail, though perhaps at a reduced capacity and functionality. For example, a smart phone should continue to be available for recording encounters even if the communication link to the nearest hospital or the cloud fails temporarily. More importantly, data that have been understood to have been committed to permanent storage should not be lost if a smart phone or micro controller board in a health kit malfunctions. We believe that standard replication and transaction management techniques in distributed data bases can handle the vast majority of such faults. We do not assume that components within the trusted computing base of the system will behave in Byzantine ways. However, the system should be able to work in conjunction with a large number of devices and with software outside the trusted computing base. (Ensuring that the trusted computing base is free of bugs will however be no easy task.)

At the storage level, the adversary can attempt to read stored data with whatever access control privileges and rights are at its disposal. It can also forge information and store it in the repositories, in the hope of making principles act as oracles to learn some critical information. Again, we do not consider an adversary being able to fill up the repositories with “junk” thus preventing genuine mHealth data from being saved on it.

At the computational level, we assume that the adversary may corrupt the integrity of *bona fide*medical data or metadata by tampering with them during processing or by linking to malicious libraries at runtime or forging information; it may also compromise confidentiality by gaining access to sensitive information by requesting permission at installation time or by uploading the information to a public server without informing the concerned user.

At the policy level and deployment level, we assume that principals and organisations involved have reasonable security policies, and that the hardware and software components employed can correctly implement these policies. Moreover, we assume that they will employ strong cryptographic techniques and reliable access control mechanisms. The challenge is to support interoperation of different organisations by ensuring information exchange amongst them while respecting the privacy/integrity policies of one another.

The security mechanisms that we explore in our design cannot address threats that arise due to the faulty working of sensors. To some extent, such problems are dealt with in the validation layer of the "sensor stack".

4.3 Tags, Labels, Authority

The DIFC framework of Myers and Liskov [ML97, ML98] allow principals to express their privacy/integrity concerns about their data by *tagging* program and data components. The tags indicate ownership of the components as well as who may legally read the data according to a desired policy. Labels are sets of tags. DIFC mechanisms track data as they flow through the system and restrict the release of information. In systems such as IFDB [SL13], data objects are immutably labelled, whereas processes reading the data get "tainted" by the labels of the data read. Very roughly, information is permitted to flow from a source s to a destination d if the label of s is contained in the label of d. Thus information may be released to the public only by processes having the lowest possible label. In Flume [KYB^{+}07], which uses DIFC framework at the level of standard OS abstractions, the focus is information declassification/endorsement as data flow through interprocess communication *endpoints*.

Declassification allows particular tags to be removed from labels, and is useful in releasing information to authorised principals or to declassify summary information that may not reveal individual sensitive data. Since it removes constraints on permitted information flow, declassification is permitted only for processes having the requisite authority on the corresponding tags. Ownership of data determines having the capability to declassify and to delegate this authority to others (and to perhaps revoke this authority later).

DIFC frameworks presented in the literature rely on principals providing tags appropriately for the various components. We believe that while this may be possible in a small system, especially one involving security-aware users, it is unrealistic for the general populace, particularly uneducated or uninformed individuals, to provide tags or to even comprehend the consequences of a security policy. Since users cannot be expected to tag their data, *we propose that data be tagged automatically and systematically*, by which we mean that tags are provided for different data fields of an encounter record by a layer within the "sensor stack". The granularity at which data are labelled in data bases may be at the level of relations (tables), records (tuples) or fields. While in [SL13], an excellent case is made for labelling data at the level of tuples, in mHealth systems we suggest that labelling may have to be at the level of fields (or more precisely, collections of fields), even though per-field labelling may involve a significant overhead. The reason for this is the following: it is not obvious who the owner of an encounter record should be. The common belief that the *patient* should be the owner of her data is not appropriate, since the encounter record may have several fields, particularly metadata fields, of which she was totally unaware and probably unconcerned. These metadata were collected as contextual evidence, and may include the unique device number of a sensor, that may be of interest to the hospital administrator or the device manufacturer, but have no relevance for the patient. Were the patient the owner of these metadata, explicit declassification would become necessary for queries related to this information. Moreover, as discussed earlier, different portions of the encounter record may be distributed across different data stores, each operating under a different privacy/integrity policy. Therefore labelling at the granularity of records is not appropriate. Preventing a blow-up in the size of tagging information is a crucial problem that we are studying.

4.4 Non-invertible, Structure-Preserving Functions

It is a mistaken belief that anonymity and privacy can be achieved by eliding identifying information from records. It is also often believed that by renaming identifiers (obfuscation), one can achieve anonymity. There have been numerous instances of data compromise due to such unjustified assumptions. It is therefore necessary to transform the data by applying functions that are difficult to invert (e.g., hash functions, one-way functions etc.). However, any query on the data, such as range queries, need to be performed on the data "going below" the transformations. Very roughly, if h is a transformation and $\oplus$ an operation on data $x_1, \ldots, x_n$, we would like to compute $\oplus(x_1, \ldots, x_n)$. However, we are not given $x_1, \ldots x_n$ but instead are presented $h(x_1) \ldots h(x_n)$. h is said to be *homomorphic* with respect to $\oplus$ if $h(\oplus(x_1, \ldots, x_n) = \oplus'(h(x_1), \ldots, h(x_n))$ for some $\oplus'$. In the kind of operations we have examined, it may not be necessary to require that the transformations be fully homomorphic. Finding weaker structure-preserving properties and transformations that allow us to perform the desired operations is a topic for future study.

The requirement of non-invertible functions also applies to the labels that we generate systematically. The tags are usually opaque strings that should not themselves reveal information. However, if they are being systematically generated, they may be created based on the kind of data with which they are associated (the field name) as well as other metadata that can be used to identify an encounter. It is therefore necessary to use one-way functions or hash functions that do not reveal much information about the inputs. The operation that one performs on tags are checking subset inclusion, which may only require weak structure preservation.

Finally, the mHealth system that we envisage involves data being shared between different administrative domains. Of course, this should be permitted when the domains agree that they will respect each other's security policies, without necessarily having to take an union of the two sets of policies. This requires checking tags that were generated in another domain, with the interpretation of the tags being understood only in that domain. We believe that the right approach to truly decentralised information flow control lies in being able to transform tags generated in one namespace to those in another namespace through a difficult-to-invert transformation, and to perform the information flow control checks in the transformed domain.

5 Conclusion

In trying to address scalability and trustworthiness issues in developing an mHealth system, we encountered the full variety of issues that distributed systems have to address:

- Communication protocols at different levels of the stack, especially within the application layer;
- Data base representation and efficient data retrieval issues;
- Consistency semantics in distributed data repositories;
- Security issues beyond encryption and authentication;
- Making systems work with legacy applications and taking into account future changes in software used.

The major system design issues involved are understanding the requirements, picking a good model and set of associated techniques, optimising them with respect to the constraints placed on the system, and finally understanding whether these design choices allow the system to operate efficiently at scale.

Among various ideas in distributed computing, some concepts stand out as pearls, using which dependable and reliable systems can be built: sequential consistency, serialisability, linearisability, atomicity, idempotent operations; store-and-forward communication, pipes for interprocess communication; public-key encryption; failure detectors, and several others. Real-world problems help us identify such key concepts from a gamut of proposals. Our still-early study of mHealth systems suggests that to this list one may add: (i) Hyper-graph data bases; (ii) graph-oriented distributed hash tables; (iii) CRDTs; and (iv) Decentralised Information Flow Control techniques.

Acknowledgements. This work has been supported by DeitY, Government of India, for the project "Foundations of Trusted and Scalable "Last-Mile' Healthcare". I wish to acknowledge fruitful discussions with Kolin Paul, David Kotz, and Aarathi Prasad, and the work of Ashwani Priyedarshi, Chandrika Bhardwaj, Aruna Bansal, Avval and Anju Kansal.

References

[ABK12] Avancha, S., Baxi, A., Kotz, D.: Privacy in mobile technology for personal healthcare. ACM Computing Surveys 45(1), 3 (2012)

[BKT01] Buneman, P., Khanna, S., Tan, W.-C.: Why and where: A characterization of data provenance. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 316–330. Springer, Heidelberg (2000)

[Bre01] Brewer, E.A.: Lessons from Giant-Scale Services. IEEE Internet Computing 5(4), 46–55 (2001)

[CPS$^+$12] Cheng, W., Ports, D.R.K., Schultz, D.A., Popic, V., Blankstein, A., Cowling, J.A., Curtis, D., Shrira, L., Liskov, B.: Abstractions for Usable Information Flow Control in Aeolus. In: USENIX Annual Tech. Conf., pp. 139–151 (2012)

[DD77] Denning, D.E., Denning, P.J.: Certification of Programs for Secure Information Flow. Commun. ACM 20(7), 504–513 (1977)

[Den76] Denning, D.E.: A Lattice Model of Secure Information Flow. Commun. ACM 19(5), 236–243 (1976)

[EKV$^+$05] Efstathopoulos, P., Krohn, M.N., Vandebogart, S., Frey, C., Ziegler, D., Kohler, E., Mazières, D., Kaashoek, M.F., Morris, R.: Labels and event processes in the Asbestos operating system. In: ACM Symp. on Operating Systems Principles, pp. 17–30 (2005)

[GMS05] Germanakos, P., Mourlas, C., Samaras, G.: A Mobile Agent Approach for Ubiquitous and Personalized eHealth Information Systems. In: Proc. of the Workshop on Personalization for e-Health of the 10th International Conf. on User Modeling, pp. 67–70 (2005)

[Ior10] Iordanov, B.: HyperGraphDB: A generalized graph database. In: Shen, H.T., Pei, J., Özsu, M.T., Zou, L., Lu, J., Ling, T.-W., Yu, G., Zhuang, Y., Shao, J. (eds.) WAIM 2010. LNCS, vol. 6185, pp. 25–36. Springer, Heidelberg (2010)

[KG14] Kansal, A., Gupta, A.: Sensor Stack on Android for mHealth Applications. Master's thesis, Department of Computer Science & Engineering, IIT Delhi (2014)

[KGPP14] Kansal, A., Gupta, A., Paul, K., Prasad, S.: mDROID - An Affordable Android based mHealth System. In: International Conf. on Health Informatics. SciTePress - Science and Technology Publications (2014)

[KK03] Kaashoek, F., Karger, D.R.: Koorde: A simple degree-optimal hash table. In: Kaashoek, M.F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, pp. 98–107. Springer, Heidelberg (2003)

[KYB$^+$07] Krohn, M.N., Yip, A., Brodsky, M.Z., Cliffer, N., Kaashoek, M.F., Kohler, E., Morris, R.: Information flow control for standard OS abstractions. In: ACM Symp. on Operating Systems Principles, pp. 321–334 (2007)

[MBN$^+$10] Mechael, P., Batavia, H., Kaonga, N., Searle, S., Kwan, A., Goldberger, A., Fu, L., Ossman, J.: Barriers and gaps affecting mhealth in low and middle income countries. In: A Policy White Paper commisioned by The mHealth Alliance (2010)

[ML97] Myers, A.C., Liskov, B.: A Decentralized Model for Information Flow Control. In: ACM Symp. on Operating Systems Principles, pp. 129–142 (1997)

[ML98] Myers, A.C., Liskov, B.: Complete, safe information flow with decentralized labels. In: Proceedings of the 1998 IEEE Symposium on Security and Privacy, pp. 186–197 (May 1998)

[OKOG12] Otieno, C.F., Kaseje, D., Ochieng, B.M., Githae, M.N.: Reliability of community health worker collected data for planning and policy in a peri-urban area of kisumu, kenya. Journal of Community Health 37, 48–53 (2012)

[PPM+13] Prasad, A., Peterson, R.A., Mare, S., Sorber, J., Paul, K., Kotz, D.: Provenance framework for mhealth. In: Fifth International Conference on Communication Systems and Networks, COMSNETS 2013, Bangalore, India, January 7-10, pp. 1–6 (2013)

[Pri11] Priyedarshi, A.: Caching and Distributed Data for Cloud-Style Computation. Master's thesis, Department of Computer Science & Engineering, IIT Delhi (2011)

[SL13] Schultz, D.A., Liskov, B.: IFDB: decentralized information flow control for databases. In: Proc. of the 8th ACM European Conf. on Computer Systems, pp. 43–56. ACM (2013)

[SMK+01] Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM 2001, pp. 149–160. ACM, New York (2001)

[SPBZ11] Shapiro, M., Preguiça, N.M., Baquero, C., Zawirski, M.: Conflict-free replicated data types. In: Défago, X., Petit, F., Villain, V. (eds.) SSS 2011. LNCS, vol. 6976, pp. 386–400. Springer, Heidelberg (2011)

[SRC84] Saltzer, J.H., Reed, D.P., Clark, D.D.: End-to-end arguments in system design. ACM Trans. Comput. Syst. 2(4), 277–288 (1984)

[Swa12] Swasthya Slate of Public health foundation of India (2012), `http://www.swasthyaslate.org`

[WHO11] WHO. mhealth: New horizons for health through mobile technologies Global Observatory for eHealth Series, 3 (2011)

[WHO12] WHO. Management of patient information: Trends and challenges in member states. Global Observatory for eHealth Series, 6 (2012)

[ZBWKM06] Zeldovich, N., Boyd-Wickizer, S., Kohler, E., Mazières, D.: Making Information Flow Explicit in HiStar. In: USENIX Conf. on Operating Systems Design and Implementation, pp. 263–278 (2006)