Scalability and Trustworthiness in mHealth Systems

Sanjiva Prasad
IIT Delhi
(joint work with K Paul, C Bhardwaj, Avval, A Kansal, A Bansal, D Kotz ..)
Outline

• mHealth Systems
• Acquisition: *Encounters, Metadata*
• Repository Organization
• Decentralized Trustworthiness
• Conclusions
Introduction
mHealth

Using smart phones to improve Healthcare

Technology Enablers:
- Ubiquity of cheaper and better smart phones with more compute power
- Cheaper, better portable sensors
- Cloud technologies
- Better communication networks and technologies
Why mHealth Systems?

Have the potential to change health care model:
• Cheaper
• Better

Adapt similar technology infrastructure for
• Personalized healthcare in developed world (monitoring patients)
• Underserved communities in developing (sharing resources)
Typical mHealth System

A Typical mHealth Scenario

Heena using Sensor-Kit to get Puja's Hb reading
Typical mHealth System

Heena using Sensor-Kit to get Puja’s Hb reading

Heena’s PDA

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System
Typical mHealth System

A Typical mHealth Scenario

Heena using Sensor-Kit to get Puja's Hb reading

Puja-Heena Encounter Data (Personal, Biometric, Medical data & Metadata)

Updating Puja-Heena Encounter Data

EMR (Hospital)

Internet

Hospital Domain

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario

Heena using Sensor-Kit to get Puja’s Hb reading

Puja-Heena Encounter Data (Personal, Biometric, Medical data & Metadata)

Hb readings from Sensor-Kit

Heena’s PDA

Updating Puja-Heena Encounter Data

EMR (Hospital)

Accessing patients’ EMR

Doctor’s PDA

Emergency Updates

Updating Patient’s EMR with new test results

Accessing patient’s EMR

Dr. Divya

Regular Monitoring

Ultrasound Laboratory PC

Doctor’s Laptop

Ultrasound Technician

Hospital Domain
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario

Heena using Sensor-Kit to get Puja’s Hb reading

Puja-Heena Encounter Data (Personal, Biometric, Medical data & Metadata)

Hb readings from Sensor-Kit

Heena’s PDA

Internet

EMR (Hospital)

Doctor’s PDA

Dr. Divya

Emergency Updates

Regular Monitoring

Accessing patient’s EMR

Accessing patients’ EMR

Updating Patient’s EMR with new test results

Ultrasound Laboratory PC

Doctor’s Laptop

Ultrasound Technician

Hospital Domain
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario

Doctor's PDA

Accessing patients’ EMR

Updating Patient’s EMR with new test results

Accessing patient’s EMR

Dr. Divya

Regular Monitoring

Ultrasound Laboratory PC

Doctor’s Laptop

Hospital Domain

Heena using Sensor-Kit to get Puja’s Hb reading

Puja-Heena Encounter Data (Personal, Biometric, Medical data & Metadata)

Hb readings from Sensor-Kit

Heena’s PDA

Typical mHealth System

EMR (Hospital)

Updating Puja-Heena Encounter Data

Accessing patients’ EMR

Emergency Updates
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System
Typical mHealth System
Typical mHealth System
Typical mHealth System

Daily Activity data from Activity Sensor

Puja's Phone

Activity Data

PHR

Internet

Updating Puja's Activity details
Typical mHealth System

Daily Activity data from Activity Sensor

Puja's Phone

Activity Data

PHR

Updating Puja’s Activity details

Internet
Typical mHealth System
Typical mHealth System
Typical mHealth System
Typical mHealth System

Daily Activity data from Activity Sensor

Puja's Phone

Activity Data

PHR

Updating Puja's Activity details

Internet
Typical mHealth System
Typical mHealth System
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Typical mHealth System

A Typical mHealth Scenario
Requirements of a usable mHealth system

- Automated data collection by health workers with minimal training
- "All-in-one" portable mHealth system
- Inexpensive for mass usage in under-developed countries
- Data representation in exchangeable format to allow usage in heterogeneous systems
The V’s of Data

Characteristics of Data generated in Health Care systems:
• Volume
• Variety
• Veracity
• (Velocity an issue in hospitals)
• Visibility-restricted (Anonymity, Privacy)
Challenges

- Mobility of people, devices, sensors
- Device trustworthiness
- Communication: Asynchrony, delay-tolerance
- Streaming data (monitoring)
- Security, integrity, authenticity of data
- Administrative domains, diverse policies
- Installed legacy in data bases, systems
- SCALABILITY
Data Acquisition and Querying
mDroid Architecture

Medical Sensors Connected to Patient → Microcontroller Board → Android Based System → Health records on Server
Medical Encounter

- **Collection of a set of data**

**And metadata**
- Using a set of sensors
  - Variety in the data: discrete, interval, continuous

- Checking/Validation of the data
- Encapsulation of data from multiple sensors
- Finally to server/cloud via smart phone
Electronic Record of an encounter

- Has data and metadata.
Metaphor: Conversations

Medical History as an on-going conversation

- A collection of records, outcome of "encounters".
- Not just the doctor-patient, but many principals involved.
- Non-humans (sensors, phones) chirp in too.
- Meta comments, asides, sub-conversations, …
- What structures? Linear? Tree? DAG?
- Causal relationship, temporal, or some other connection?
Metadata Matters!

Contextual Evidence

• Who, Whose, When, Where, using What,…
• (Why? How?)
• Provides some assurance of quality of data
  • Confidence in the data
  • Can use metadata to audit
• cf. Provenance
Various uses of Metadata

• Is the Health worker doing her job honestly, does she follow protocols correctly? *(Admin)*
• Do others in the area also have the same symptoms *(Public Health Research)*
• Is there a correlation with weather and environmental conditions? *(Diagnosis)*
• Is this device working properly? *(Operational)*
• Is this brand of sensor good? *(Commercial)*
Queries
On medical data
On medical data and metadata
Purely on metadata

Need a well-designed query language
  SQL?  SparQL?  Xquery?  …
How should the data be collected & stored?
XML? RDF?  …
Storage and Retrieval Issues
An encounter record may be distributed.
Heterogeneity Rules, ok?!

Data Bases in different Admin Domains

- How does one add *metadata* in *systematically, and system-wide*?
- Various parties: patients, health workers, doctors, hospitals, researchers, administrators, regulators,…
- Hospitals unlikely to migrate their data.
- Relational formats are poorly suited
- NoSQL? MongoDB? XML databases? Key-value stores?
- Federated, heterogeneous, flexible.
HyperGraphical Forms

Flexibility in representation

- Decentralized, Expandability
- N-ary relations
- Higher-order and meta-relations
- Graph-oriented storage
- Graph traversals, Path queries
- Locality in query processing
Distributed Data Base Issues

Atomicity, concurrency, fault-tolerance, …

When does an encounter complete?
  When the data is send to smart phone?
  When the data is committed to cloud?
Where is the “record”?  

Consistency vs Availability vs Partition
Consistency Semantics

Eventual Consistency

Monotonicity assumption
No information is deleted
- May be deprecated, ignored

Commutative operations: orthogonality
Conflict-free operations
Convergent Replicated Data Types
No complex concurrency control required
Trustworthiness

Keep your word
Decentralized Trustworthiness

Privacy and Integrity

Data
  - at rest,
  - in transit,
  - being processed in a computing device

Properties to be ensured:
  - End-to-end confidentiality
  - End-to-end integrity

Across different administrative domains
Adversary

A general concept capturing computational limitations

- Device failures
- Network faults: messages lost, duplicated, delayed, corrupted
- Active eavesdropper can inject fabricate nonces, data, hash functions, ...
- Attempt to downgrade information, link to malicious libraries
- Attempt unauthorized access, installation-time
- Dolev-Yao to compute/resource-limited adversary
Building for Trustworthiness

The usual array of solutions

- Data at rest: Access control
- Data in Motion: Cryptography
- Data being processed: Information Flow Control
- Hardware failures: Redundancy
- System Failures: Replication, Transactions
- Network faults: Delay- and Fault-tolerance

Need to be integrated systematically
Ownership and Access

Even within a single administrative domain:
- Policies about data access to principals
  - Access privileges
  - Ownership of data
  - Special Capabilities
    - Revoking privileges
    - Allotting privileges

In Single vs Multiple Domains
MYTH:
Entire *Encounter record* is owned by *Patient*!
MYTH:
Entire *Encounter record* is owned by *Patient*!
MYTH:
Entire *Encounter record* is owned by *Patient*!
MYTH:
Entire *Encounter record* is owned by *Patient*!
MYTH:
Entire *Encounter record* is owned by *Patient!*

It’s difficult to even specify the needs of principals.
MYTH:
Entire *Encounter record* is owned by *Patient*!

It’s difficult to even specify the needs of principals.
MYTH:
Entire *Encounter record* is owned by *Patient*!

It’s difficult to even specify the needs of principals.
Sharing Data

Data sharing between two administrative domains

• “MoU”
  • Enforce privacy policy of one domain in another (and vice versa)
  • Restrict data release to third parties

• Understanding of how policies will be enforced in the second domain.
  • Cannot take simple union of policies
Information Flow Control

Security Classes and Policies

- Security Class Lattice [Denning 1977]
  - No Read-Up, No Write-Down
  - Relaxed Non-Interference
- Decentralized IFC: Myers\&Liskov:
  - User-supplied labels (tags)
- Adapted for OS-level, process level.
- Static vs dynamic analysis techniques
Anonymity is not privacy

Meaning of anonymity in graph-structured data is very different from relational datasets.

Basic topological structure of the social graph can act as identifier.

An active attack in which adversary deliberately introduces random edges in the social network so that the resulting subgraph can be recognized. Even after all information about identities has been removed from the network.
SIFT: Tagging Data Systematically

- DIFC work assumes data tagged by users.
  - Not practical for naïve users
  - User unaware of many metadata
    (e.g. sensor Id, fault history)
- Fine-grained tagging: Tags automatically added in the SensorStack:
  - Tags should not reveal information, yet be amenable to processing
  - Tag operations may need to peek under the hash function
Storing Tags Systematically

How and Where should tags be stored?

• Finding appropriate Data structures and IR techniques
• Store them where they are needed.
• Exploit spatial and temporal **locality** on tag queries

What does the marriage of medical database with the repository for tags & labels look like?
Implementation challenges

- How large is the label space?
  - Is fine-grained labelling feasible?
  - Does it scale?
- How opaque can one keep the tags?
- Obfuscation does not really give security
- One-way functions: computationally hard to invert
- Weaker structure-preserving / topology-preserving functions (cf homomorphic encryption)
- DHTs (Chord/Koorde do so for location)
Conclusion
In Summary

mHealth systems encounter the full spectrum of issues:

- Designing an appropriate Communication Stack
- Efficient Distributed Data Bases, Query, Retrieval
- Ensuring Consistency
- Security: beyond authentication and encryption
- Legacy of the installed base and anticipating the future
- Building a suitable trusted computing base
The new pearls?

- Hypergraph Data Bases and the use of Programming Language Concepts in Distributed Data Base Query
- Graph-oriented DHTs for efficient distributed information retrieval
- Convergent Replicated Data Types (CRDTs): minimal concurrency control mechanisms
- Decentralized Information Flow Control
Questions?
Thank you!

- We are looking for graduate students to work in the areas of distributed systems, security and program analysis