Secure and Scalable RESTful Health Data Exchange

Gerald Beuchelt, Rob Dingwell, Andrew Gregorowicz, Harry Sleeper
www.projecthdata.org
Background

• EHR technologies are currently at the heart of the national U.S. Health Care debate
  – Promise of significantly improved efficiencies and cost savings
  – Improvements in the quality of care

• EHR systems have been around since the 1960s
  – Massachusetts General Hospital MUMPS and Intermountain HELP system
  – Some EHR systems use MUMPS today: Veterans Administration’s VistA
  – There are over 100 “modern” EHR implementations

• Yet, adoption rates in the general medical community have been very low as of 2009:
  – Less than 11% of U.S. Hospitals have comprehensive EHR systems
  – Less than 18% of physicians have access to EHR systems

• Deployed EHR systems are often non-interoperable
Introducing hData

- **hData** is a new approach to EHR standards
  - Strict separation of content and format: medical community defines the content, technical community defines the format
  - Machine and human readability of source XML is important

- **Collection of linked, but standalone XML documents**
  - MUST provide schema, so docs can be validated
  - Goal is to have small XML documents
HRF Abstract Structure

hData Record Format (HRF)

Root Document

Section (Name, Type of Documents, Path)

Section Documents

References

Section

Section Documents

Section

Section Documents
Web Representation

- HRF Structure maps naturally to URLs
  - Base URL identifies the record
  - Section paths map naturally to relative URLs
  - Section documents are of Content Type application/xml

- Section URLs resolve to Atom feeds
  - Default feed: contains section documents
  - Alternative feed: contains child sections

http://example.com/hdata/patient1234/adversereactions/allergies/1.xml

<table>
<thead>
<tr>
<th>Record Identifier</th>
<th>Section(s) Path</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolves into root document or user interface</td>
<td>Resolves into Atom feed of documents or sections</td>
<td>Content Type application/xml</td>
</tr>
</tbody>
</table>
RESTful API

- All entities are subject to RESTful operations (GET, PUT, POST, DELETE)
  - Entire hData Record
  - Sections or child sections
  - Individual section documents

- Some operations may not be defined on a resource
  - For example: the root document may only be accessed by GET
  - Only limited processing instructions are specified

- Benefits
  - Easy to implement – compatible with wide range of tools
  - Internet scalability – up to 100 Millions of users
  - Result: faster development cycle, more innovation
Departure from Tradition

• Traditional health records are a snapshot in time
  – Paper based: copy of current records are shipped
  – CDA-based EHRs: electronic representation of “point-in-time” records
  – Discrete information transmission through HL7 messages introduces additional complexities

• An hData Record is a living document
  – Once an hData Record resource location is known, services can subscribe to content feed
  – Automatic, timely updates and changes based on open standards
  – Service consumers can copy an entire hData Record information for “point-in-time” documentation purposes

• Subscription access can be cut off
  – For example: Patient changes specialist – no further access for that specialist to the patient’s hData Record is necessary
Access Control, Identity Management, and Privacy

• Basic access control available today through modular design
  – Already better current situation
  – But: Coarse granularity – section documents are the unit of protection
  – May cause section proliferation (e.g., separation of behavioral records in separate tree)

• Privacy and Access Management
  – Looking at Kantara User Managed Authorization, a four-legged OAuth protection scheme
  – Focus on protection of PII and HIPAA compliant profile

• Future requirements
  – Minimally: Section document based granularity
  – Ideally: XML node based access control
  – Other ideas: signed section documents
Scenario: Near Real Time Updates

Patient registers with his hData discovery system.
Patient sees his PCP during a Regular checkup – everything is ok.
EHR System Registration

Patient sees his PCP during a Regular checkup – everything is ok

PCP hData EHR system gets registered with the patient hData discovery service.
Emergency

Sees emergency room after a sports accident.

Patient → ER Surgeon → Patient
EHR Discovery

Patient

Sees emergency room after a sports accident.

ER Surgeon

ER Surgeon’s EHR System registers as a new provider EHR system and discovers existing EHR systems.
Data Retrieval and Subscription

Patient

Sees emergency room after a sports accident.

ER Surgeon

Gets the relevant data from the PCP EHR system. Also subscribes to PCP EHR system.

PCP
Performing Services

Patient

Performs procedure and prescribes medication.

ER Surgeon

Updates record with new data.

PCP
Discovery Service Check

PCP EHR system checks discovery service for updates.
Patient discovery system notifies of new EHR system.
Local Data Update

PCP system contacts the new EHR system to obtain new data and subscribe to ER Surgeon’s EHR system.
Local Data Update

Patient

ER Surgeon

Updates record with new data

PCP
Follow-up Visit

Sees his PCP again after The accident. PCP prescribes some additional medication.
Near Real Time Update

ER Surgeon EHR system gets medication update in near real-time, since it is subscribed to patient’s EHR record on PCP system.
Near Real Time Notification

ER Surgeon’s EHR system can warn the patient and the PCP about potential adverse effects in near real-time.
Scenario: Provider Change

Patient

Patient decides to change surgeons and notifies discovery service to block the ER surgeon’s subscription access.

ER Surgeon

PCP
Policy Update

Patient

ER Surgeon

PCP’s EHR system check discovery service for updates.
Discovery service instructs EHR system to cancel ER Surgeon’s subscription access.
Subscription Access Blocked

ER Surgeon’s EHR system gets blocked at the next update attempt.
Preliminary Architecture

• RESTful Discovery
  – Goal: Simple URL/identifier to “hook” into the federation
  – Using OASIS XRD 1.0 for creating provider specific XRD to discover actors in the medical federation
  – hData Discovery and Authorization Service will need to allow user to determine specific profiles

• RESTful Authorization
  – IETF OAuth 1.0a (including session fixation fix) as candidate
Advanced OAuth

• Basic OAuth is too simple
  – User interaction required, not concept of centralized Authorization Manager (=PDP)

• Alternative: Kantara UMA
  – 4-legged scenario
  – Allows pre-authorization and (limited) policy management
  – Currently under development
SCAP and hData

• Two types of hData systems
  – hData Discovery and Authorization Service
    “Federation Hub”
  – hData EHR/PHR systems
    “Federation Member”

• No guarantee that hData systems are run by full-time staff
  – hData DAS can – technically – be operated by patient
  – hData EHR system at doctor’s offices, labs, etc. → many small and medium businesses
**SCAP Benefits**

- Compliance with regulations
  - HIPAA SR 164.312(a)(1) Access Control mapped to SCAP through NIST 800-53
  - Automation critical to typical operators
- Action item
  - Map hData HIT regulations and requirements to SCAP
  - Compile XCCDF profile and SCAP content for hData DAS and hData EHR systems
Putting it together

C32 et al. is like a James Joyce novel: beautiful, dense, hard to read

hData is like the graphic novel edition: it maintains the richness of characters and the storyline, but is much more accessible

Parallel approach offers alternatives to speed interoperability
hData Deployment and Integration
Resources

• hData home page: http://www.projecthdata.org/
  – Current versions of the hData Specifications
    • hData Record Format
    • NQF-35 hData Content Profile
  – L32 information
• Feedback: talk@projecthdata.org