

# hData - A Simple Approach to Health Data Exchange

---

Gerald Beuchelt, Rob Dingwell, Andy Gregorowicz, Harry Sleeper

<http://projecthdata.org/>

The MITRE Corporation

202 Burlington Rd.

Bedford, MA 01730

*Health data interoperability issues limit the expected benefits of Electronic Health Record (EHR) systems. Ideally, the medical history of a patient is recorded in a set of digital continuity of care documents which are securely available to the patient and their care providers on demand. The history of electronic health data standards includes multiple standards organizations, differing goals, and ongoing efforts to reconcile the various specifications. Existing standards define a format that is too complex for exchanging health data effectively. We propose hData, a simple XML-based framework to describe health information. hData addresses the complexities of the current HL7 Clinical Document Architecture (CDA). hData is an XML design that can be completely validated by modern XML editors and is explicitly designed for extensibility to address future health information exchange needs. hData applies established best practices for XML document architectures to the health domain, thereby facilitating interoperability, increasing software developer productivity, and thus reducing the cost for creating and maintaining EHR technologies.*

---

## 1 Background

In Section 1, we discuss the motivation for addressing the problem space of health data exchange formats. We briefly revisit the history of the work in the field and outline problems with the currently accepted approach. In Section 2, we introduce a modern XML best practices approach to organizing health data for electronic consumption. Section 3 outlines how our approach can be aligned with the existing work in the field of electronic health records. Finally, in Section 5, we conclude with a brief review.

## 1.1 Electronic Health Data Exchange

Electronic documentation of health care data is currently at the heart of the U.S. national discussion on healthcare reform. While there is no universally accepted definition of an Electronic Health Record (EHR), we follow the common approach of referring to the entirety of the electronic data about a single patient as the complete EHR, while the data stored in a single system is referred to as the Electronic Medical Record (EMR) [1] [2]. Electronic Health Record Systems (EHR Systems) have existed since the 1960s with the introduction of MUMPS [3] and have benefited from the information technology advancements of the last 40 years. However, health data exchange interoperability and other usability issues have plagued system-wide adoption [4] and have thus limited the expected benefits. As of 2009, adoption rates in the U.S. have been as low as 11% for hospitals [5], while only 17% of all U.S. physicians have access to an EHR System [6].

For achieving quality outcomes and economic efficiencies, summary of care information plays a special role. Ideally, the entire relevant medical history of a patient is recorded in a set of summary of care documents securely available to patients and their clinicians on demand. This way, the entire care team gets the same exact and complete picture of the patient's health data, without costly repeated examinations, duplicate lab tests, and partially reported conditions or results (reported to one physician, but absent from another physician's records).

## 1.2 The Path to HITSP C32

Health Level Seven (HL7) is a health standards organization whose work focused on health data standards by creating the clinical document architecture (CDA) [7]. The CDA was created with complete coverage of edge cases in mind: using the CDA, one can expect to address nearly all documentation needs in any health care system. Consequently, this approach made the schema extremely flexible but overly complex, hard to implement in an interoperable design, and difficult to manage. In the meantime, the Massachusetts Medical Society and others created a simpler continuity of care record (CCR), not based on the CDA. Eventually, another standards organization, ASTM International, adopted the CCR as its proposed continuity of care record standard [8].

HL7 reconciled its standards with ASTM by taking the data elements found in the CCR and encoding them in the CDA, with the resulting standard being called the Continuity of Care Document (CCD). As part of the U.S. national initiative in Health Information Technology (HIT), the American National Standards Institute (ANSI) and its affiliated Health Information Technology Standards Panel (HITSP) recommended that the CCD be used as an input standard for creating a national continuity of care standard. The result was initially published as the HITSP Construct 32 standard (C32) [9]. The module content and supporting vocabularies were recently migrated to HITSP documents C80 [10] and C83 [11]. As a result, the latest HITSP C32 is essentially only a reference to C80 and C83. The entire document suite has been significantly expanded and has grown more complex.

### 1.3 Criticism of the CDA and C32

There are several shortfalls in the CDA, C32, and related standards. Most of these were experienced firsthand when we implemented “Laika,” an open source C32 compliance testing tool set (<http://projectlaika.org>). Four of the key issues are described below.

- **Repeated use of overly abstract data structures:** The HL7 CDA defines a number of very generic objects that are used to represent information in a given document. Differing information, such as medications and conditions, are represented using the same XML elements with very subtle changes in their nesting and attributes. This makes a CDA document difficult to process.
- **Underspecified implementation, including lack of a normative schema:** While there is an XML schema for the HL7 CDA, a final schema does not exist for the HITSP C32 or other CDA-based documents due to their use of attributes for selecting templates. Thus, defining schemas for these documents is impossible. As a result, CDA-based constructs such as HITSP C32 cannot be automatically validated by XML parsers; standard object mapping tools, such as XML Beans or JAXB, cannot be used.
- **Ambiguous data types:** Data can be represented in multiple ways in a CDA document. Consumers of CDA documents must, therefore, write software that handles any of the numerous permutations of these data types. This leads to bloated software, or more likely, software that does not implement the full specification and experiences interoperability problems when it receives data in an unexpected format.
- **Steep and long learning curve:** Mastery of the CDA and its many specifications and constructs takes an experienced software engineer many months to achieve. Once learned, it is very cumbersome to employ in robust software applications and services. These difficulties drive up the cost and time to develop and maintain health care software, thus reducing the pace of innovation.

While we are not aware of any widespread operational deployment of CDA or C32 for health information exchanges, work on these standards has created useful medical domain expertise in the health industry. HITSP has recognized the complexity of the existing CDA-based standards and has completed an effort to “streamline” the standards and the documentation on how to use them [12]. At the heart of this effort lies the definition of HITSP Capabilities and Service Collaborations. The Capabilities are essentially profiles of existing HITSP Constructs which then map to requirements of the American Recovery and Reinvestment Act (ARRA) [13] (see <http://hitsp.org/default.aspx?show=library> for an overview of the HITSP Constructs and [14] for the EHR Centric Interoperability Specification). The HITSP architecture approach after the Spring 2009 Tiger Team review is described in [15]. While this document reordering may provide some help, exchanging continuity of care information will still take place in the same overly complex format.

Additionally, the entire existing HITSP framework does not always deliver a comprehensive, interoperable set of specifications, thus exacerbating interoperability problems. For example, the latest revision of the HITSP C19 Construct [16] references Integrating the Health Enterprise (IHE) IT

Infrastructure Technical Framework (ITI-TF) Volume 2 [17] and relies on ITI-TF 2, section 3.40, “Provide X-User Assertion” for exchanging user attributes. Section 3.40 is a very loose profile of using restricted SAML 2.0 assertions [18] with WS-Security [19] and WS-Trust 1.3 [20]. It is unclear why WS-Trust was chosen over the SAML 2.0 protocol, especially since WS-Trust does not define use case profiles, processing rules, or static conformance rules. IHE failed to provide complete use case profiles and processing rules in its specification. Omissions like these invite vendor-specific interpretations of the underlying standards and—in the absence of coordinated, point-to-point interoperability certification testing—will lead invariably to non-interoperable solutions.

## 2 Introducing hData

As an alternative to the CDA framework, we propose hData, a simple XML framework for the creation, storage, and exchange of health data. The hData specification [21] contains three components:

1. hData Record Format (HRF): The HRF describes an abstract architecture of how data is stored in multiple XML documents and organized in a hierarchy. It also contains a concrete schema for the HRF metadata. Records conforming to the HRF are called hData Records (HDRs).
2. HRF Serialization: Within EHR Systems, hData Records may be persisted in different ways. The hData specification describes a portable scheme to create an archive of the individual documents that make up the HDR.
3. hData RESTful API (HRA): When the HRF is represented as a web resource, this RESTful specification allows for modification of section documents, creation of new data, record transport, and management of the entire record through a simple RESTful Web API.

Beyond the above technical specification, hData uses hData Content Profiles (HCP) to specify the actual content included in a particular hData record. It is important to note that the HRF determines the format of the record but does not determine the medical data that needs to be contained within an HDR. The HCPs determine what data must be contained within an HDR for a particular purpose. As such, a given HDR will always be conformant with the HRF and may satisfy one or more HCPs.

The data elements defined in the NQF-35 hData Content Profile [22] address the immediate issues of continuity of care interoperability, but its scope goes well beyond this application. By including all of the National Quality Foundation’s (NQF) 35 data elements (NQF-35) derived from the NQF Health Information Technology Expert Panel (HITEP) [23], the NQF-35 HCP is suitable as a general purpose electronic health record format. Any hData implementation must minimally support the NQF-35 hData Content Profile. Please visit <http://projecthdata.org/> for more information on the hData Content Profiles and the NQF-35 HCP specification.

Separating the hData Content Profiles from the technical container and communication specification serves the interests of both the health care and technical communities: the HCPs should be defined and managed by medical domain experts, while the data organization issues can be better addressed by data management experts. By separating the problem areas and creating two distinct specification sets, the

respective communities and their standardization organizations can employ their own expertise and follow their own time lines.

While the hData technical specification and the NQF-35 hData Content Profile make up the core of the hData framework, additional hData Content Profiles can be created to extend the functionality of the HRF (e.g., a lab report or an immunization record HCP could be easily added). Going forward, we will develop other HCPs that cover specific use cases. In addition, in Section 4, we briefly discuss possible access control, identity management, and privacy management extensions for hData that use the hData RESTful API.

## 2.1 hData Overview

The hData Record Format (HRF) follows the approach taken by the Open Document Format (ODF) [24] and other modern XML file formats: at the core of the document is a “root” document containing metadata describing the actual medical data documents, which are located within a hierarchy of sections. These individual XML documents are referred to as “section documents” and are located within a section. Any given section can only contain section-documents of one type or other sections. These sections can easily be represented as a file folder hierarchy on disk or within a ZIP file, or as web resources.

The hData Record Format was created with extensibility in mind. Since we do not expect to be able to address all potential use cases with a single HCP, hData can be extended by defining new sections for additional XML documents to deliver additional functionality with almost no limitations to the format of the extensions. While highly desirable, we do not expect that all consumers of hData will be required or, indeed, will be capable of parsing all documents, so extensions must be marked as mandatory or optional.

Finally, the NQF-35 hData Content Profile aims to enhance EHR data quality by enforcing strict rules on the non-narrative parts of the record to allow automated machine processing. C32 and related formats often use narrative fragments within fields that are intended for machine consumption, thus breaking interoperability. For example, dates can be specified through descriptive terms such as “a week ago”. While there is a requirement to capture the fact that a date is ambiguous, resorting to unstructured text creates significant interoperability issues and makes machine parsing of EHRs unnecessarily complex. By restricting common data types to well-established type definitions, hData lowers the interoperability barrier and simplifies the creation of health care software.

## 2.2 Record Organization

Data in the hData Record Format is stored in a set of linked standalone XML documents. Each data point in a patient’s record is captured in an independent standalone XML document. The collection of these documents, along with some organizational metadata, constitutes an hData Record that conforms to the HRF specification. There is no constraint on the XML schema used in each of these individual documents: existing data (e.g., legacy data, machine generated lab results, etc.) can be integrated into an HDR without loss of fidelity by adding a section for the legacy data XML documents. Non-XML data

can be wrapped either through simple transforms where possible, or by encoding the data in a form suitable for XML storage (e.g., by using BASE64 encoding for binary data).

These independent documents are linked and organized in a hierarchy: a “root” document contains metadata about the structure and content of the collection. In particular, the following metadata resides in the root document:

- **Unique document identifier, creation date, and modification date:** The document identifier should not be tied to a particular resource or document location, but must instead guarantee uniqueness over all other hData documents. The identifier should not be overloaded with any semantics. Valid choices for the document identifier could be a UUID [25], or a URL that uses the root of the EHR System operator and a unique identifier for the document itself.
- **Extensions:** Since hData is extensible, all contained types of section documents, identified by an XML namespace, must be registered in the extensions section. Once the universal resource indicator (URI) for a section document is registered in the `Extensions` node of the root document, this type of section document can be used in any `Section` node. To assist hData parsing, extensions are marked mandatory or optional. Optional extensions can be ignored. If mandatory extensions cannot be parsed, the system must at least notify the user that critical portions of the document will not be processed.
- **Section definitions:** Sections can contain data documents of a single type or other sections (sub-sections). The type of section document, identified by a URI, must in turn be registered in the extensions node of the root document. The hData Content Profile defines a standard section layout that can be extended by the implementer. Note that there can be many sections that may contain the same type of documents, (i.e., the relationship between section document types to sections is one-to-many). It is helpful to think of sections as “directory folders” containing individual XML documents (“files”) and other sections (“folders”). Figure 1 illustrates the hData Record Format structure.

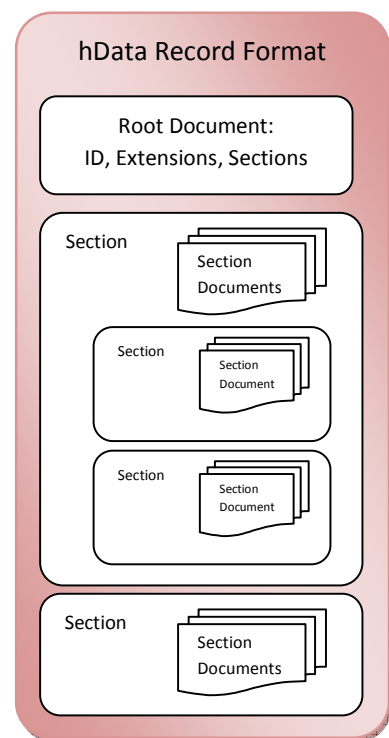


Figure 1: HRF Structure

There is no restriction on the content of the section, with the exception that the “section documents” must be expressed in XML and have a type registered as an extension in the root document. Figure 2 contains a very simple example of a root document.

```

<root xmlns="http://projecthdata.org/hdata/schemas/2009/06/core">
  <documentId>c64e620d-f648-4531-9703-14b37afefc2c</documentId>
  <version>0.1</version>
  <created>2009-07-12-04:00</created>
  <lastModified>2009-07-12-04:00</lastModified>
  <extensions>
    <extension requirement="mandatory">
      http://projecthdata.org/hdata/schemas/core/2009/09/patientinformation
    </extension>
    <extension requirement="mandatory">urn:empty</extension>
    <extension requirement="mandatory">
      http://projecthdata.org/hdata/schemas/2009/06/allergies
    </extension>
  </extensions>
  <sections>
    <section
      typeId="http://projecthdata.org/hdata/schemas/2009/09/patientinformation"
      name="Patient Information" path="patientinformation"/>
    <section
      typeId="urn:empty"
      name="Adverse Reactions" path="adversereactions">
      <section
        typeId="http://projecthdata.org/hdata/schemas/2009/06/allergies"
        name="Allergies" path="allergies"/>
      </section>
    </section>
  </sections>
</root>

```

Figure 2: Root Document Example

## 2.3 Network Access to hData Records

The data represented by an hData Record should be simple to exchange from one EHR System to another. The hData Record Format lends itself ideally for RESTful applications: the same hierarchy that can be represented in a file folder structure can transfer directly to a URL hierarchy. With this representation in mind, we define a RESTful Application Programming Interface (API) to edit the patient hData Record through HTTP at the section and section document level. This approach is much more efficient than the complex IHE XDS protocol [26] currently proposed for sharing CDA-based health data.

By using the HTTP GET verb, any part of an HRF document can be read: the root document is directly accessible through the base URL of the HRF. The root document contains all necessary metadata to access the medical information contained in the HRF document, including a list of all mandatory and optional document types a parser must implement. Each section is accessible as a sub-resource, identified by its path-segment, which in turn is used to compute the absolute URL to the section resource. A GET operation on the URL of a section (or sub-section) resource returns an Atom 1.0 feed of all documents or a feed of its sub-sections. The returned results can be filtered by using HTTP headers, such as TE (Transfer Extension) headers [27], or other appropriate query parameters. For example, a GET on /immunizations with If-Modified-Since set to a specific date, would only return immunizations that have been administered since a given date.

Modifications of the HDR are implemented analogously. New section documents can be added or modified through PUT or POST operations; DELETE works in the same fashion, although we recommend well-defined auditing processes when deleting data from an HDR. Again, HTTP headers can be used: ETags are highly useful for PUTs. For example, a new allergy can be added to the document by PUTting

an allergy document into the /adversereactions/allergies section in the document hierarchy.

Existing EHR systems (such as the U.S. Veterans Administration's VistA) can be retroactively equipped with hData capabilities without having to re-architect the underlying EHR system.

## 3 HITSP Alignment

### 3.1 C32 Interoperability and the L32

In order to maintain interoperability with existing EHR implementations that use the C32, we also introduce a mechanism to map between hData and a tightly profiled version of the HITSP C32 standard, called the "Lightweight C32." The "Lightweight C32," or L32 for short, is a specification that bridges between CDA-based architectures and hData. There are two "modes" for L32: native and compatibility. Both remove some of the ambiguity found in the HITSP standard. A native mode L32 document can be easily validated against the new L32 XML Schema, which is not compliant with the C32 and CDA standards. This is due to the use of xsi:type attributes in the L32 that facilitate the creation of a more concrete schema. For L32 documents in compatibility mode, these xsi:type attributes can be added via XSLT to provide compatibility with the C32 standard. Conversion between native and compatibility mode is performed through a simple XSLT transform.

L32 provides a simpler path to generate a document that will be C32 conformant, and satisfies the requirements of current and proposed EHR legislation in the U.S. L32 is currently under development; more information is available on the Project hData home page at <http://projecthdata.org/>.

Mapping between hData and L32 is achieved through a simple XSLT/XProc process. This transformation is necessary since C32 or its descendants are required in regulated U.S. EHR Systems. Using this approach, new systems can fully focus on leveraging the simplicity and precision of hData, thus eliminating the need to maintain the complex organizational knowledge and skills required for consuming or producing C32 or other CDA-based constructs. Existing systems can leverage the constrained C32 profile and the hData conversion tools to ensure that their continuity of care documents are interoperable. Since hData is capable of providing all functionality for a CDA-based document system through extensibility, the hData format allows for a natural evolution away from the CDA-based document formats.

The L32 alignment approach should not be mistaken for a solution in and of itself. The L32 maintains the cumbersome CDA legacy, and cannot address any of the problems deriving from its monolithic form. L32 is intended to ease the transition from HITSP C32, not to address all of the requirements for a comprehensive EHR health data standard. As such, L32 is limited to the continuity of care sections of the NQF-35 hData Content Profile.



## 3.2 HITSP Harmonization Framework

While hData introduces a new approach to the data format and replaces significant portions of the existing CDA architecture, it still fits conceptually into the HITSP architectural framework [15]. The extensive existing work on defining code systems, data standards, data dictionaries, data exchange content, and use case scenarios can be leveraged with hData through its simple extension mechanism. hData Content Profiles, if hData is eventually adopted, will rely heavily on the medical domain knowledge that HITSP and its members have been working on successfully for many years.

In fact, individual HITSP constructs (including CDA-based documents such as the C32) and other legacy EHR formats, could be included in their own section within an HDR. While this is not the intent of hData, this approach offers an easy migration path away from CDA-based health data exchange.

## 4 The Road Ahead

The hData technical specification described so far is not sufficient to build a fully featured EHR system or to fully enable emerging EHR technologies [28]. We intentionally restricted the scope of the hData specification to the data architecture to avoid the complex interdependencies seen in the CDA and the specifications that build on the CDA. Going forward, hData needs to address the challenges of the complex interactions of the health care industry as well as the requirements of health regulation. In this section, we briefly outline the near-term roadmap for hData by addressing three crucial problems:

- **Access Control:** Access to an HDR must sometimes be restricted to a subset of the data contained in the HDR. The HRF offers a natural fine granularity for access control at the level of individual section documents. Access to individual section documents, as well as entire sections, can be restricted through Access Control Lists (ACLs) or section path-based patterns.
- **Identity and Privacy Management:** By using access control features, access authorization can be made identity aware, i.e., access to individual sections or section documents is granted based on the identity of the requesting entity. Building on such an identity management framework, access to HDRs can be managed to preserve privacy along the lines of HIPAA and ARRA, but also other generally accepted privacy guidelines such as the OECD Privacy Principles [29].
- **Discovery and Patient Empowerment:** Ultimately, we see a strong need for empowering patients who should have easy access to their complete EHR. In addition, every patient should also be enabled to share their complete or self-selected EHR data with anyone they choose.

### 4.1 Access Control and Identity and Privacy Management

The hData specification does not include any access management components. We have consciously decided to focus on data modeling first, and to create hData Content Profiles for additional functionality later. At the same time, the overall architecture is well-suited for fine-grained access control that allows for privacy and addresses confidentiality needs:

- Section documents are the basic units of data storage. Since these documents are self contained, any access management system that uses this document architecture already has a granularity that cannot be achieved with any CDA-based document format.
- The hierarchical organization of the sections also makes it easy to grant access to sections or sub-trees of sections by hierarchy. This enables a variety of interesting applications, such as anonymization through blocking access to the `/patientinformation` section.
- The web resource-based representation of hData Records lends itself to defining access policies based on URL patterns, which is readily supported today by several vendors and open source software projects.

There are a variety of potential ways to define identity, privacy, and access management profiles for hData, including Access Control Lists (ACLs), policy agents, or even Simple Object Access Protocol- (SOAP-) based identity web services. At this time we are focusing on demonstrating a RESTful protection scheme, which works with the hData RESTful API. It builds on the “ProtectServe” protocol that has been presented by Eve Maler et al. [30]. ProtectServe has been submitted to the User-Managed Access Working Group of the Kantara Initiative [31] for public review and standardization.

## 4.2 Enabling Patient-Centric Electronic Health Records

As indicated in Section 1, a complete EHR is the collection of all individual EHRs and EMRs across all health service providers that hold health data about a patient. Within the health community, there are many EHR Systems operated by different actors, such as health providers, government entities, insurance companies, and others. Currently, patients have very limited electronic access to their digital health data stored by other actors, and even less active control over who can access their health data. Since hData implements RESTful patterns, access to hData Records for patients can easily be implemented by giving patients access to their data over the internet. Simple stylesheets can be used to display the contained data in a human-readable form.

In addition, hData web resources can be made discoverable through web-centric protocols such as the proposed XRD 1.0 protocol [32]. By using discovery mechanisms, patients can effectively link their HDRs in different EHR Systems and create a more complete picture of their health data. By using a ProtectServe-like authorization scheme, the patient is empowered to authorize access to their records based on the identity of the requestor.

A combination of XRD discovery and ProtectServe authorization management is illustrated in a simple use case in Figure 3 that illustrates the following steps:

0. Prior to any interaction, the patient or service provider configures the Discovery and Authorization Service (DAS). This includes registering existing HDR endpoints and, optionally, pre-configuring HDR default access policies.
1. When requesting the medical services of a new provider (e.g., a specialist), the patient interacts with the EHR System of the specialist by pointing the provider’s EHR System to the patient’s DAS.

2. At the DAS, the patient selects the applicable HDR endpoints that he wants to share with the specialist. This step is necessary, since the patient may choose not to share some HDRs (e.g., mental health data). At this stage, it is important for the DAS to present the patient with reasonable default privacy choices, and to warn the patient of potential risks when disclosing information beyond the default selections.
3. A custom XRD is generated for the specialist and sent to the specialist's EHR System.
4. The specialist's EHR System can now contact the HDR endpoints contained in the custom XRD and initiate a ProtectServe session and provide access to the patient's authorized HDRs.

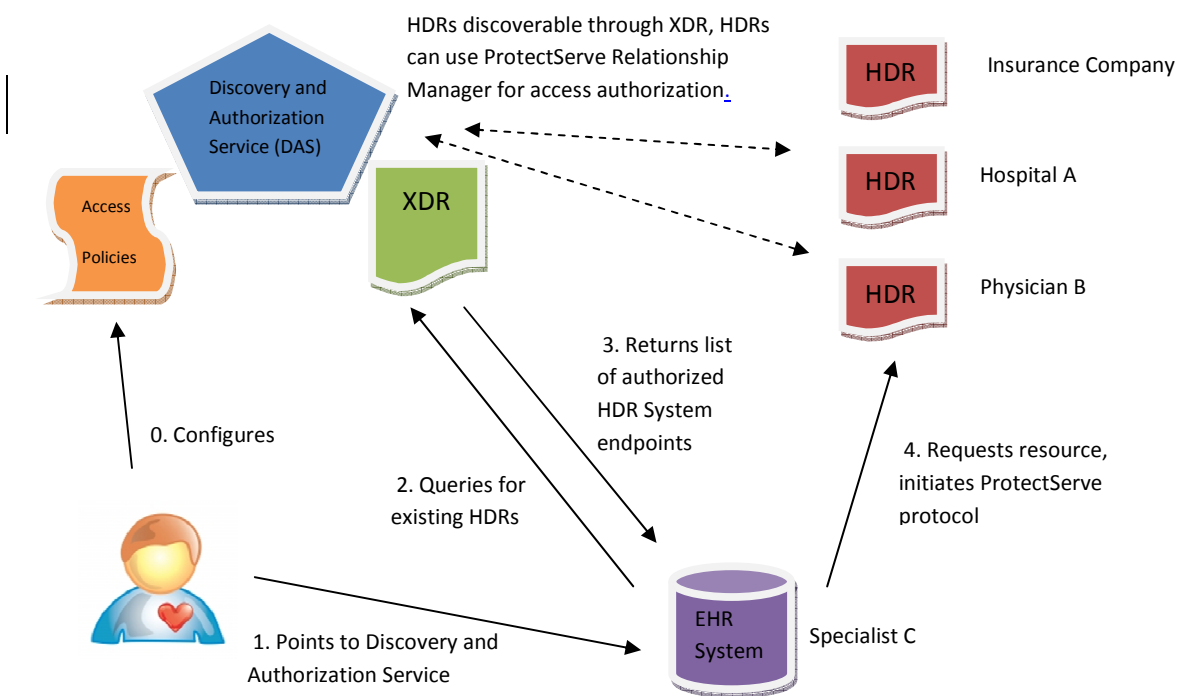


Figure 3: Notional hData use case

Since all protocols for this exchange are completely open and no additional knowledge beyond the location of the HDR endpoints is necessary, a patient can choose to implement their own DAS, making the system truly patient-centric. At the same time, we expect that either existing actors in the medical community such as health providers, government entities, insurance companies, or emerging service providers similar to Google Health or Microsoft HealthVault will offer DAS to patients.

## 5 Conclusions

hData is a new approach to address the current complexities of electronic health data exchange. The hData design avoids many of the problems of the CDA outlined in Section 1.3:

- **Repeated use of overly abstract data structures:** The structures defined in the HRF are extremely simple, and the use of hierarchical storage for individual section and section documents is well-understood in the software developer community.
- **Underspecified implementation, including lack of a normative schema:** The hData technical specification comes with a small normative W3C XML Schema for the metadata root document. The NQF-35 hData Content Profile defines a normative W3C XML Schema for the National Quality Foundation's 35 data elements. By providing clearly defined XML schemas, the hData framework guarantees automated XML tool support which simplifies software development.
- **Ambiguous data types:** All hData constructs have a single representation that can easily be processed by machines and humans.
- **Steep and long learning curve:** hData can be learned and used in a software system in days, not weeks or months, reducing the cost and time to develop health care software. This will accelerate the innovation of EHR technologies.

In addition, hData is designed so that implementation is simple, fast, and cheap. Through its simple extensibility model, hData can address the needs of summary of care documentation and can also be profiled as a comprehensive EHR data container. hData is highly portable through its simple serialization format and its RESTful API. hData is extremely flexible, can be validated completely by modern XML parsers, and is capable of addressing both current and future health data exchange needs.

By going beyond the outdated monolithic document architecture, hData enables new EHR technology features such as fine-grained updates of individual portions of a patient's record, fine-grained access control on all health data, and simple integration with existing and emerging identity and access control management systems.

## 6 Acknowledgements

We would like to thank Beth Halley, Joy Keeler, Michael Macasek, and Mary Pulvermacher for their insightful comments.

## 7 Bibliography

- [1] D. Garets and M. Davis, "Electronic Medical Records vs. Electronic Health Records: Yes, There Is a Difference," 2006.
- [2] EMR and HIPAA Forum. What's the difference between EMR and EHR?. [Online].  
[http://emrandhipaa.com/wiki/What%27s\\_the\\_difference\\_between\\_EMR\\_and\\_EHR%3F](http://emrandhipaa.com/wiki/What%27s_the_difference_between_EMR_and_EHR%3F)
- [3] Wikipedia. [Online]. <http://en.wikipedia.org/wiki/MUMPS>
- [4] C. Weimar, "Electronic Health Care Advances, Physician Frustration Grows," *Physician Executive Journal*, Mar. 2009.
- [5] M. E. Schneider, "Adoption of EHRs by U.S hospitals is low," *OB GYN News*, May 2009.
- [6] (2009, Feb.) Modern Healthcare. [Online].  
<http://www.modernhealthcare.com/article/20090206/REG/302069989>
- [7] Health Layer 7, "HL7 Clinical Document Architecture, Release 2.0," 2004.
- [8] ASTM, "Continuity of Care Record," ASTM E2369-05.
- [9] HITSP Construct, "Summary Documents Using HL7 Continuity of Care Document (CCD) Component," HITSP/C32, 2007-2009.
- [10] HITSP Construct, "Clinical Document and Message Terminology Component," HITSP/C80, 2009.
- [11] HITSP Construct, "DA Content Modules Component," HITSP/C83, 2008.
- [12] HITSP, "Report from the Tiger Teams," Presentation HITSP 09 N 412, 2009.
- [13] 111th Congress, "American Recovery and Reinvestment Act of 2009," 2009.
- [14] HITSP Construct, "EHR-Centric Interoperability Specification," HITSP/IS107, 2009.
- [15] HITSP Construct, "Exchange Architecture and Harmonization Framework Technical Note," HITSP/TN904, 2009.

- [16] HITSP Construct, "Entity Identity Assertion Component," HITSP/C19, 2009.
- [17] IHE International, "IT Infrastructure Technical Framework Volume 2," ITI-TF-2, 2008.
- [18] OASIS Standard, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0," 2005.
- [19] OASIS Standard, "Web Services Security: SAML Token Profile 1.1," 2006.
- [20] OASIS Standard. (2007, Mar.) WS-Trust 1.3. [Online]. <http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html>
- [21] G. Beuchelt, R. Dingwell, A. Gregorowicz, and H. Sleeper, "hData Specification," MITRE Corporation, 2009.
- [22] G. Beuchelt, R. Dingwell, A. Gregorowicz, and H. Sleeper, "NQF-35 hData Content Profile," MITRE Corporation, 2009.
- [23] National Quality Foundation, "Health Information Technology Expert Panel Report: Recommended Common Data Types and Prioritized Performance Measures for Electronic Healthcare Information Systems," 2008.
- [24] OASIS Standard, "Open Document Format Version 1.0," 2005.
- [25] P. Leach, M. Mealing, and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace," RFC 4122, 2005.
- [26] IHE International, "IT Infrastructure Technical Framework Volume 1," ITI-TF1, 2008.
- [27] R. Fielding, et al., "Hypertext Transfer Protocol -- HTTP/1.1," IETF RFC 2616, 1999.
- [28] D. C. Kibbe and B. Klepper. (2009, May) The Health Care Blog. [Online]. [http://www.thehealthcareblog.com/the\\_health\\_care\\_blog/2009/05/an-open-letter-to-the-new-national-coordinator-for-health-it-part-3-certification-as-the-elephant-in.html](http://www.thehealthcareblog.com/the_health_care_blog/2009/05/an-open-letter-to-the-new-national-coordinator-for-health-it-part-3-certification-as-the-elephant-in.html)
- [29] Organisation for Economic Co-operation and Development, "OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data," OECD Recommendation, 1980.
- [30] E. Maler. Pushing String. [Online]. <http://www.xmlgrrl.com/blog/categories/protectserve/>
- [31] Kantara Initiative. Kantara Initiative. [Online]. <http://kantarainitiative.org/>

[32] E. Hammer-Lahev, "XRD 1.0," 2009.