Model Formulation ■

The PING Personally Controlled Electronic Medical Record System: Technical Architecture

WILLIAM W. SIMONS, MS, KENNETH D. MANDL, MD, MPH, ISAAC S. KOHANE, MD, PHD

Abstract Despite progress in creating standardized clinical data models and interapplication protocols, the goal of creating a lifelong health care record remains mired in the pragmatics of interinstitutional competition, concerns about privacy and unnecessary disclosure, and the lack of a nationwide system for authenticating and authorizing access to medical information. The authors describe the architecture of a personally controlled health care record system, PING, that is not institutionally bound, is a free and open source, and meets the policy requirements that the authors have previously identified for health care delivery and population-wide research.

■ **J Am Med Inform Assoc.** 2005;12:47–54. DOI 10.1197/jamia.M1592.

The recently released strategic framework for the National Health Information Infrastructure calls for the creation of personal health records.[1] Personally *controlled* health records[2,3] represent a subset of the possible implementations of personal health records in that they are designed to exist outside the administrative structures of any particular health care institution.[1,4,5] Most existing personal health record systems are institution- or company-specific repositories for health information.[6] We have designed and implemented an instance of an interoperable, personally controlled health record that we call the Personal Internetworked Notary and Guardian (PING).[7] PING has been designed to capture the usage, control, and range of policies that we have previously described, many of which have been adopted as part of the national agenda.[4]

It is our goal in this paper to describe an architecture that we believe would enable the implementation of the national goals and to discuss the motivation behind each architectural decision. The architecture that we describe enables individuals to have an integrated record over time and across institutions to securely store and access that record and to have fine-grained control in delegating access to portions of that record. The PING architecture will support the evolution of underlying data models such as the HL7 RIM and LOINC[8] as well as specific applications that can be built to serve local needs, such as online scheduling of appointments and patient–doctor communications. We want to describe the architecture for three reasons: first, to use the architectural description to fully specify the desiderata that we had previously alluded to only elliptically and thereby provide grounds for very specific agreements or disagreements with our design choices; second, to motivate others to take our free open-source code base [PING software and documentation are available under the Gnu Lesser General Public License (Gnu LGPL) at http://ping.chip.org] and implement PING applications and contribute their enhancements to the PING code base; third, to encourage others to provide similar transparency and disclosure in the designs of other personal health records. While it has become increasingly accepted that the largest hurdles in implementing a society-wide software solution are not technical ones, the architectures of these solutions can contribute to the difficulty or ease of addressing the societal roadblocks. To date, even when consumer groups and health care institutions have been motivated, they have been stymied by poor interoperability of software and vendor-locked, nonextensible software that does not conform to standards.[4] Furthermore, as societal concerns are likely to change, adopted architectures should allow flexibility of locus of control and capabilities. Again, not all architectures or their implementation are equally amenable to the necessary flexibility.

We include below, a brief scenario to serve as a motivation of our architectural decisions. The scenario is annotated with the relevant architectural components. These components are illustrated in Figure 1 and described in detail in the following section.

Affiliations of the authors: Children's Hospital Informatics Program (WWS, KDM, ISK), Division of Emergency Medicine (WWS, KDM), and Division of Endocrinology (ISK), Children's Hospital Boston, Boston, MA; Harvard Medical School, Boston, MA (KDM, ISK).

Supported by the National Library of Medicine through contracts N01-LM-3-3515 and N01-LM-9-3536.

The authors thank Matvey B. Palchuk, MD, MS, for his work on developing the original PING clinical document data model and access control model. They are grateful to Ulrich Sax, PhD, and Lucas Jordan for their ongoing effort in defining the new PING record and clinical document data models.

Correspondence and reprints: William W. Simons, MS, Children's Hospital Informatics Program, Enders 1, 300 Longwood Avenue, Boston, MA 02115; e-mail: <william.simons@childrens.harvard.edu>.

Received for publication: 03/26/04; accepted for publication: 09/21/04.

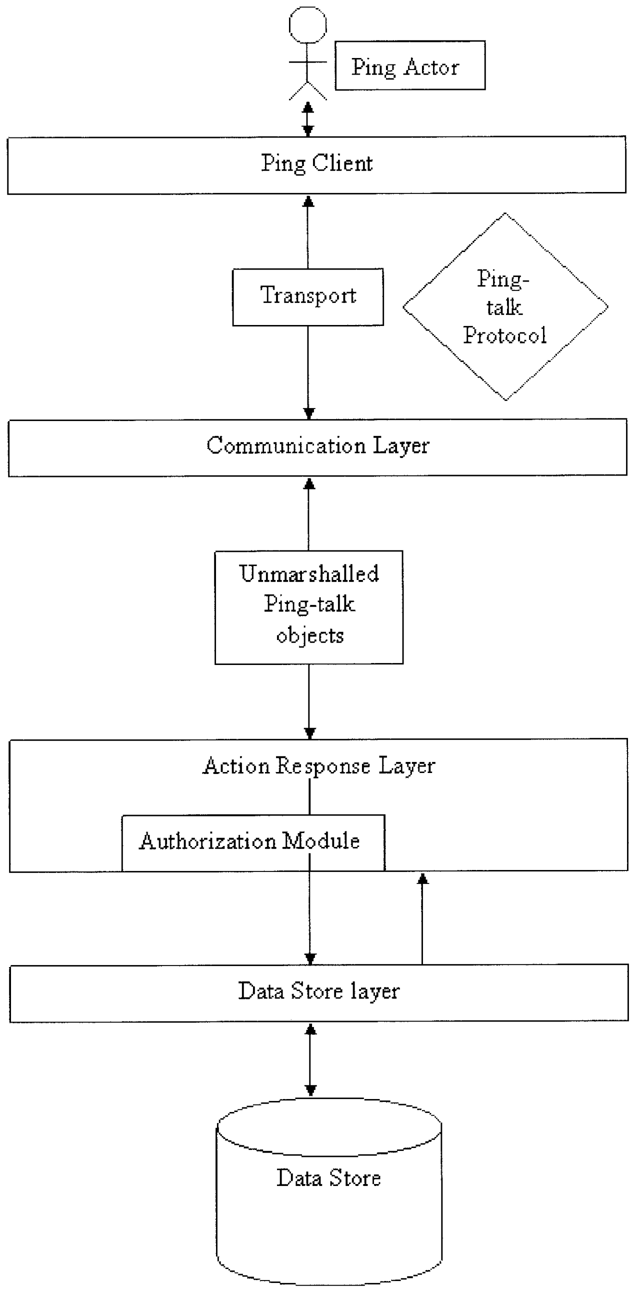

Figure 1. PING layers.

Scenario

Jill [PING Actor] is visiting her primary care physician, Dr. First [PING Actor] who is conveniently located a half mile down the dirt road. She just has had a yearly repeat mammogram at the Women's Medical Center (WMC) 20 miles away where she "loaded up" her PING record with the images and reports via the WMC's patient portal [PING Client]. Dr. First reviews the mammogram and compares it with the last two, also obtained at WMC. There are no changes. As Jill is currently taking L-thyroxine for hypothyroidism, he reviews the dose, uploaded [PINGTalk, Communication Layer] into the PING record from one of the two regional pharmacy benefits management companies and notes that she is taking ten times the dose that he had prescribed. Before he has a chance to be alarmed, he sees an annotation entered by Jill in which she notes that the dose is incorrect and that she has notified her local pharmacy. At the end of the visit, Dr. First informs Jill that given the strong family history of ovarian and breast cancer, she might want to consider genetic testing. After further discussion, Jill wants to think about it some more, and so Dr. First uploads the consent and also an order form that she can bring to, or electronically share with, a regional commercial laboratory where the blood sample would be obtained. He also asks her if she could provide him read access [Authorization Module] to the result and she immediately agrees. However, when he asks her if she would like to make her PING record available [by providing read access to a PING polling program (PING Client)] to an anonymized population-wide query of patients at high risk of breast cancer, she expresses reservations.

Five weeks later, the genetic test comes back with equivocal results. It does not seem that Jill has any of the known cancer-associated mutations in the genes that they tested, but there is a novel, possibly clinically irrelevant mutation on one gene. Dr. First immediately refers her to a genetic counselor at the New Cancer Center (just across the street from WMC but a direct competitor of WMC). Jill asks her family members whether any of them have had the same test. One aunt has done so, and Jill provides her with a cover sheet that she can use to fax [PING Client] a copy of that test into the family history section of her PING record. Jill meets with the genetic counselor who was able to review, before the meeting with Jill's explicit permission, the primary documentation of the WMC studies, Jill's genetic test, her aunt's test, and all the notes and tests performed by Dr. First.

PING Overview

For those who are unfamiliar with PING, it is a system designed as a fully distributed electronic medical record in which patients have control over who can read, write, or modify components of their records. It is not designed to be the primary record of the health care system but a comprehensive compilation of all medical data longitudinally across the patient's history. PING allows a patient to make and maintain his or her own electronically collated copies of his or her record encrypted in a storage site of his or her choosing. Storage sites can range from server farms to individual Internet server provider (e.g., America Online) accounts. Access, authentication, and authorization all occur on one of several available PING servers, which are also responsible for encryption of the record. To date, PING systems have shown early promise through prototype applications at Children's Hospital Boston and in primary care clinics and unconventional venues for health care (e.g., supermarkets) in Canada. The latter have been developed independently by developers using the aforementioned public open-source code base (see Validation section). Specific applications for these pilots include maintaining continuity after discharge from the emergency department and integrating the pharmacist and dietitian into the documented care of the patient. Currently, the largest PING development effort is to provide an integrated view of patient populations in the New England area under routine and disaster conditions.[9]

Background

The PING architecture is not the first model to address the issues involved in allowing patient access and participation in the development of their medical records. CareGroup, a multihospital and physician practice network in eastern Massachusetts, has developed an application called

PatientSite[10] that seeks to address some of the aspects of personal health records. PatientSite allows patients to view portions of their medical record and annotate those portions; it can also be used by patients for nonurgent communication with their providers. As described below, the architecture of PING allows these actions and others by adopting patient ownership of their records as a primary goal. To this end, the PING server allows not only administrative-level authorization of actions, but also patient-level authorization. Another distinction is that PING is designed to allow new actions to be added to the server. Some of these actions allow and promote provider input directly into the patients' PING records in standard formats.

Other groups have used storage schemes similar to that of PING. In the virtual patient record system of Malamateniou and Vassilacopoulos,[11] eXtensible Markup Language (XML) is used as the primary means of representing patient records. Rather than relying on a single set of XML schemas, the PING data model is kept as generic as possible, as discussed below, to allow emerging document standards to be stored and the authorization on those documents to be easily maintained.

PING Architecture

Our primary goal in designing PING was to explicitly factor the architecture into modular pieces that could be replaced by various parties so long as there is adherence to specified core protocols. The purpose of this paper is to make this factoring explicit so that collaborators can knowledgeably add to or substitute in PING components. As diagrammed in Figure 1, the components include a PING client, a communication layer, a pluggable action-response layer, an authorization module, and a data store layer. The figure also illustrates the dataflow among these components. Each component is now described and motivated below.

The PING Actor

The PING actor is a user (or agent[7]) of the PING system. Each actor in the system is assigned various attributes when his or her record is created. These attributes include roles, group memberships, and proxy representations. A role in PING is associated with certain privileges such as the ability to create records, read records, and add or update elements in records. For instance, an actor logged in with a "patient" role may have the ability to read and update his or her own record but not be able to create new records for others. An important note about privileges is that even though someone may have the privilege to perform an action in some records, that action may be restricted by patient-defined access policies in any particular record. In the case in which an actor has multiple potential roles (a doctor can also be a patient), the actor must designate the appropriate role upon login.

Group memberships do not have privileges associated with them. Their usefulness resides in the ability of actors in the system to set access policies on their own records to allow or forbid certain actions to members of a particular group. For instance, an actor might want to specify a policy in his or her record that allows all providers in the allergy department at his or her health clinic to add documents classified as allergy documents. This can easily be accomplished if all the providers have a group membership that identifies them as part of that department.

Finally, when an actor A is acting as a proxy for another actor B, all access policies on a record that are relevant to actor B are now relevant to actor A. This means that even though doctor A does not have any access policies on a patient's record that allow an update, since there are policies that allow doctor B to update, they would allow A to update as a proxy. Access decisions are discussed in more detail in the section describing the authorization module.

The PING Client

A PING client is any software process adherent to the PINGTalk protocol (a variant of Web services) and that interacts with the PING system via a PING server. Examples of PING clients include a Java application that provides a patient with a browsable view of his or her entire record and control of read/write/update permissions, a fax server that allows images of paper documents to be faxed into the PING record (upon authentication and authorization), and a polling application that reads a population of PING records. The implementation language or platform for the client is immaterial so long as the PING client can communicate with the PING server via the standard PINGTalk protocol. A PING client can be a software agent with any function so long as PINGTalk is adhered to and the authentication and authorization components of the protocol are observed.

The PING Server

The PING server is the point of access for all PING clients. It is responsible for ensuring proper authenticating, locating, and decrypting the PING records (which are often stored in sites different from the PING server) and honoring the authorization policies that are encoded in the particular patient's PING record. There is typically more than one PING server deployed even in a single system for the purpose of redundancy and load distribution (although load-balancing software has not been currently implemented). In addition, PING servers from other health care systems can access any PING record as long as they present the proper credentials. The PING server is composed of many interacting modules and layers (Fig. 1). These modules and layers are discussed below, along with a description of the communication protocol that clients use to contact the server.

The PINGTalk Protocol

PINGTalk is a communication protocol, based on XML messages described in a World Wide Web Consortium (W3C) XML schema, and a request-response interaction between a client and a server. The PINGTalk protocol is kept generic so that it can be easily expanded as new actions and results of those actions are defined and added to deployed PING servers. The generic PINGTalk request contains an optional session ticket that identifies the actor making the request, an action wrapper that has a description of the action that the client wants the server to perform, and all the necessary information that the server needs to process the action. Examples of this information include credentials for an authentication action, patient records to be created, query strings describing records to be read, and other information relevant to the type of action.

The server responds to an action with a response that has a structure similar to that of the request but with some slight differences. Instead of an action wrapper, the response contains a result wrapper that has a description of the result being returned and the result itself (specific to the action requested). The response also includes a status message indicating whether the action was performed successfully. Both request and response structures include attributes that identify themselves, the PINGTalk schema version that defines their structure and the time of issuance. The XML schema for the PINGTalk protocol is available online.[12]

In the PING reference implementation, we have defined six actions and their corresponding results. The actions include creating a new record, reading an existing record, authenticating to the PING server, querying for a list of records, adding a document to a record, and updating a document within a record. The XML schemas for these actions are available online.[13] Also available is an example of the request/response messages for an authentication action.[14]

The Communication Layer

The communication layer is responsible for communication over various Internet protocols and the unmarshalling and marshalling of PINGTalk messages. That is, when a request is made to the server, the communication layer extracts the XML contained in the request, converts it into a usable program object, and passes it along to the action response layer. Once the action response layer has returned the appropriate object representing the XML response, the communication layer marshals the object into the protocol's response that is sent back to the requesting party.

The reference implementation of the communication layer is handled by a Java 2 Enterprise Edition (J2EE)–compliant Servlet. The Servlet technology provides the ability to send and receive PINGTalk messages over the Secure Hyper Text Transfer Protocol (HTTPS)[15] communication protocol. The advantages to this include inherent encryption of the PINGTalk messages through the use of HTTPS and easy manipulation of the request through the existing Java Servlet application programming interface.

Although we have chosen Java Servlets for the PING reference implementation, other developers can implement the communication layer within other application frameworks and maintain interoperability with other PING servers and clients as long as adherence to the PINGTalk protocol is maintained.

The Action Response Layer

The action response layer contains the logic for processing registered actions and maintains information about actor sessions. Actions are registered during start up when the controller module of the layer is initialized. During the initialization phase, the controller reads a configuration file that contains the information necessary to instantiate modules that are registered to handle a particular action. The information used in instantiation typically includes definitions of data stores, authorization engines, and the type identification of actions that the module will handle. When an action request is sent to the server and passed on to the action response controller, the controller delegates the processing of the action to the registered module. The modules are responsible for enforcing their own security policies and returning a result consistent with the action. Since the modules are responsible for their own security, it is essential that the server administrator understand and trust what each module is doing. With this modular approach, the PING server can be used in any Web services environment (not just specific to personally controlled records) because the logic of responding to actions is contained in these fully pluggable modules. In our reference setup of PING for personally controlled health records, we include modules for authenticating, creating new records, adding new documents to existing records, updating existing documents in records, reading a record, and querying for sets of records. Each of these modules uses the data store and authorization engine discussed in further detail below. Upon receipt of an action, the reference modules extract the relevant information and construct authorization requests for the authorization engine. If the engine permits the request, the modules construct a result based on the type of action and information that they can retrieve from the data store.

The final feature of this layer is a pluggable action extension mechanism. The mechanism is pluggable in the same manner as the action module is in that extensions are registered during the server initialization phase. Also like the action modules, extensions register for particular action events. The difference in the behavior of the extension is that it executes its task after the action module has finished, and the execution is asynchronous with the request. That is, a response is sent to the request in parallel with the execution of the extension process. To date, simple extensions have been used to report on the status of the server but more complicated extensions are being planned. One possible extension would replicate actions to a backup PING server.

The Authorization Module

The authorization module contains the logic for determining whether an actor is allowed to perform a particular action on a particular record or record element. Authorization for an action can be based on the policies of the action responder processing the action, the policies of the record (if any) that the action is being performed on, or a combination of both. To process these authorization requests, the module must be initialized in such a way that it can locate the relevant policies. In the case of locating responder policies, the module is initialized with a list of files, of which each contains one or more policies. In the case of locating individual record policies, the module is initialized with the data store in which the records are located. A typical action responder will have two instantiations of the authorization module (one to enforce the responder policies and one to enforce the record policies) and be coded to combine the results in a meaningful way.

The PING reference authorization module is based on the Organization for the Advancement of Structured Information Standards (OASIS) Extensible Access Control Markup Language (XACML)[16] specification and uses Sun Microsystems's open-source XACML library to implement the necessary functionality. A OASIS standard, XACML, that describes both a policy language and an access control decision request/response language (both encoded in XML).[17] The policy language provides the ability to describe detailed and complex access control based on the actor making the request, the action being taken, and the resource to which the action will be applied.

A major advantage of this access control model is that it is not based on a fixed hierarchy of access levels. This is especially important in a personally controlled health record such as PING in which different record owners may have different preferences concerning which individuals and groups will be allowed to view their data. That is, we explicitly reject the notion that there is a single, natural, or even fixed personal hierarchy of privacy. Depending on the idiosyncratic concerns of the individual, different items will have different privacy policies that differ from those of others and even from those at a different time in that individual's history. For example, HIV status may be of lesser concern to some individuals than the hospital at which they obtain their care. At other stages of their life, all but their resuscitation orders may be considered private by that individual.

Another advantage to using the XACML policy model is that it enables the enforcement of policies at a fine-grained level. That is, policies can be written that apply to individual documents within the record, not just on the record itself. For instance, the result of reading a record for which access is denied to certain documents will appear as if those documents do not exist.

Although the above module was designed primarily for personally controlled health records, it is possible to use it to restrict access to other types of targets in which there is no fixed hierarchy of access levels. Similarly, there are other models that may apply to personally controlled health records. One such model is Role-Based Access Control (RBAC).[18] Although PING is not based on a fixed hierarchy, RBAC relies on it. Also, in RBAC, the administrators of the system decide which roles get which privileges (and therefore access). In the PING model, administrators still assign roles to actors and create server side policies that grant corresponding privileges, but record owners can still block access. It is possible that, if a sufficient hierarchy of roles were devised, this model could be used by a PING implementation. However, such an implementation would limit an important aspect of PING: user control.

There are two other common access control models, Discretionary Access Control (DAC)[19] and Mandatory Access Control (MAC).[20] In DAC, the owner of a particular object (a record in our case) dictates the access control on that object. That is, a record owner could grant any level of access, including write access, to his or her record to any actor regardless of which roles that actor has. In MAC, the system controls the access rights on all objects (records) and the actors have no control. In PING, that would mean that an actor with a role that has the privileges to read a record could read any record simply because he or she has that role. The disadvantages of these models are clearly their lack of flexibility in privilege granting. The PING model, however, combines the best of both without the drawbacks.

The Data Store Layer

The data store layer is responsible for housing all the records that the server needs access to. The layer was designed in a pluggable fashion such that a number of different types of stores can be configured for the server to use. Interaction with a store that is not currently supported by the system can be accomplished using the PING API. The API contains a Java interface that defines the method calls that the PING server uses when communicating with the back-end store. By writing a Java class that implements the PING store interface with methods that perform the correct operations, a developer can enable the use of any back-end store.

There are many possibilities for a suitable data storage implementation for PING records, each with its own advantages and disadvantages. Perhaps the simplest implementation is to store all the records as a single, encrypted XML Binary Large Object (BLOB). The advantage of this scheme lies in its simplicity and its inherent privacy preservation (i.e. individual records are not identifiable without decrypting the entire store). The disadvantages of this model lie in the performance overhead incurred in decrypting a large BLOB each time that any action needs to be performed on any datum on any patient record. Another implementation that maintains the advantage of security and privacy involves maintaining a single, encrypted record BLOB for each user. The BLOB is indexed by a one-way hash (e.g., SHA-1) of the user's name. This allows faster look up of individual records, while still protecting the information contained within them. However, performance is still an issue when querying the store for records that contain specific information. One way to compensate for this is to create indices for queries that are commonly performed. For instance, during off-peak hours, the server would search through the records to identify those that belong to patients with a particular syndrome and store a list of those records in memory or as another encrypted BLOB. Users of the index would need to be made aware that the index was only valid up to the time of its creation.

A further step in improving performance of a store with encrypted records is to split each individual patient record into multiple BLOBs, e.g., one BLOB per clinical category. This model further increases performance while also further decreasing privacy because the mere existence of a BLOB in a particular category may be too revealing about the corresponding record.

The reference implementation of PING uses a data store that is an encrypted XML file store that follows the model of one file per user. That is, all the records are stored as files that contain digitally encrypted XML records. Other types of data stores could include relational databases and Lightweight Directory Access Protocol (LDAP)–compliant directories and meta-directories. All the models described above can be implemented in any of these back-end stores. However, there are other models specific to databases and directories that use the benefits of such systems. These benefits include preexisting indices (for data that are not encrypted), optimized search, existing query languages, load balancing, and automated backup and distribution of data.

The Data Model

The data model for a personally controlled health record in PING is designed to be as generic as possible and yet still be capable of representing a collection of specific types of medical documents. To this end, we have adopted a philosophy of wrapping externally defined documents in an XML structure that provides information about how to interpret the documents' contents. Therefore, a personally controlled record in PING is merely a collection of documents, each with meta-data that describe how that document is classified within the record, the type of document that it is, a brief description of the document's contents, and its creation and modification time.

The PING reference implementation has some predefined document types that will be used in the personally controlled records. Some of these documents are defined by the PING team, and others have been defined by others in collaboration with the HL7 Clinical Document Architecture Committee. The PING team has currently defined a document that contains the record owner's authentication credentials[21] and a document that contains information about the owner's actor attributes[22] (such as roles, group membership, and proxy information). Document types that represent various clinical documents for emergency department visits, clinical reports, laboratory results, and medications have been developed by HL7 and others.

Our motivation for this meta-data scheme is primarily that we will not be reliant on the incomplete, emerging standards but will be able to store documents produced by them and maintain useful information for processing them. We will also maintain copies of the current versions of these document descriptions so that, in the event that they change, we will be more easily able to migrate the older versions of documents to the new standard.

Validation by Example

The Canadian National Research Council is adopting PING as a model for regional, provincial, and national deployment of personally controlled health records. In April 2004, a group of developers, physicians, and policy makers collaborated to implement a functioning technical and organizational prototype of a personally controlled record system for Canadian citizens with diabetes called CitizenHealth. The system includes an application at a service bureau in which patients can register and create their PING records, transfer of laboratory data from the local hospital, transfer of current prescription information from the local pharmacy, physician and nutritionist note entries, use of voice-over-IP technology for patients to enter their daily step count, and a Web portal to view and annotate the information in their record. The critical insight gained from this exercise is that the PING architecture and code base was adequately stable and well documented such that a team of programmers in another country was able, in just four weeks, to develop six new open-source applications based on a PING server and built on the PING libraries. The success of the prototype demonstrates that our model is valuable for technology diffusion through open-source development combined with maintenance of standards for interoperability. We recognize that such prototype tests are not equivalent to full-scale deployments and do not address the larger questions of social acceptability and use. However, they do answer the first-order questions of adaptability of the architecture for a variety of applications in different social/national environments. Furthermore, the goal of this paper is to describe the architectural decisions made for PING and to encourage critical feedback from the medical informatics and larger community.

Discussion

An overarching goal for the evolving National Health Information Infrastructure (NHII) must be a robust platform for longitudinal medical records that are integrated across sites of care. The architecture that we propose is intended to support a model of a national lifelong medical record system. Our vision is an ambitious one—a nationwide, personally controlled, ubiquitous health record.

Technical Limitations

The system as currently deployed has technical limitations that we hope to address through continued development locally and by the open-source community and the emerging community around the NHII.

One specific technical limitation that the current system faces is the lack of distributed storage. Efforts are in progress to identify existing distributed storage systems and evaluate their usefulness in the PING system. Once suitable storage systems are identified, they can be easily integrated into PING by replacing the reference data store module.

Another deficiency in the current system is a robust backup and fail-over system for both the data store and the server itself. Again, a suitable distributed data storage technology will have the properties of data robustness and built-in fail-over. The issue of fail-over for the server will need to be addressed through newly developed code. A potential solution could be an extension of the action response module that replicates all transactions to a second (or any number) of PING servers deployed as "slaves."

PING's architecture does not specifically address synchronization with data providers (e.g., hospitals, laboratories), although we hope that our Web services–based approach will provide a platform from which we can develop proper synchronization applications.

Although the PING data model is capable of parsing and storing any XML-based document and the meta-data describing that document, this is not the same as being able to semantically process that document. The existing PING server and PING clients are unable to glean the meaning of arbitrary, unknown document types. We are in the process of developing a set of object identifiers that can be used to tag data elements as having particular meanings (e.g., a patient's last name) regardless of where those elements appear in a document. This process does not solve the general problem but will perhaps alleviate it in cases in which data providers are willing to tag their data but are unable, or unwilling, to format it in a PING standard manner. Notwithstanding, to the degree that the semantics of the data model standards such as HL7's Clinical Document Architecture are specified with increasing precision, the general problem (as for other personal health records) becomes substantially more tractable.

We have not addressed the issue of federating identities for patients who own multiple records on the same or different servers. To address this, we will continue to monitor the progress of groups such as the Liberty Alliance[23] and implement the concepts that they develop and promote. Another identity challenge that PING will face is the assignment of roles/groups/proxies and the corresponding privileges. Although it is not our current goal to solve this issue, the reference authorization module and its use of XACML-based policies for the server will aid in the process of assigning privileges to existing roles. The definition of those roles/groups/proxies is an institutional issue that PING will leverage once it is solved.

Finally, an ongoing challenge, although not a limitation, is for the PING team to be aware of emerging standards in all relevant areas and to be capable of adopting the new technologies and specifications.

Technology Diffusion

Of course, the main limitation of our approach is the enormous task required to push technology diffusion and societal acceptance. While we propose that the development proceed through an open-source model, ultimately, it is patients who must demand and use the software. There is a chicken and egg problem. PING will be the most effective and likely to be adopted if ubiquitous. Initial implementations will not be able to rely on ubiquity as a motivator, which is why early successful implementations motivated and funded by longer range planners is necessary. Recent emphasis by the Secretary of Health and Human Services on personal health records will be helpful in focusing the support of consumer groups on this problem. Furthermore, the increasing demand of patients to have copies of their medical record[24] and the increasing awareness of patients regarding modern arcana, such as the multiplicity of genetic tests for asymptomatic testing[25–27] in which the medical system has not developed concomitant expertise, is likely to drive the demand for direct access and control of personal medical data.

Legal

There are legal issues to be elaborated that are beyond the scope of this paper. Among the desiderata to be considered: The PING system would likely not be a covered entity under the Health Insurance Portability and Accountability Act (HIPAA) nor are the patients themselves who will use PING to compile and manage their personally controlled records/information. To the extent that covered health plans or providers adopt a version of PING for their *own recordkeeping purposes*, the records that they create and maintain in-house using their version of PING will be subject to HIPAA's protections. Health information that these entities print or download from an individual's PING records will become subject to HIPAA once it is "received" by the covered entity. If a public health official or outside researcher seeks access to, or disclosure of, an individual's protected health information from a HIPAA-covered entity, the HIPAA privacy regulation would govern. To be clear, PING cannot ensure a greater degree of accountability beyond the limits of its reach: the point of authorized release of patient information to another party. Beyond that point, HIPAA's and other regulatory protections hold. Precisely because PING places the initial disclosure under patient control, it makes explicit to the receiving party the traceable responsibility of this initial data release and the implications under HIPAA.

Conclusion

Some organizations are beginning to grant patients electronic access to a longitudinal view of their hospital-based records. Although this approach is very encouraging, it fails to solve the problem that a patient's medical records are generally fragmented across treatment sites, posing an obstacle to clinical medicine, research, and public health efforts. A system such as PING, however, thrives in the setting of such institutional efforts because data are mobilized from legacy systems and prepared for patient consumption. PING, although still a work in progress, addresses issues of portability, security, and access control primarily for data originating in institutional, office, and laboratory records. Although there are myriad issues raised by patients reading their own records,[28] we believe that, for the foreseeable future in the United States, it is only by leveraging the patients' right to have copies of their own records that we can hope to achieve effective and lifelong continuity of the medical record across all health care institutions. Since the approach of creating personally controlled health records, which we have advocated for a decade,[29] is now at the very core of the larger NHII conversation, we hope that this paper serves to help focus that conversation on the specific technical issues required to achieve the vision.

References ■

1. Thompson TG, Brailer DJ. The Decade of Health Information Technology: Delivering Consumer-centric and Information-Rich Health Care. Available at: http://www.hsrnet.net/nhii/materials/strategic_framework.pdf. Accessed Aug 24, 2004.
2. The Personal Health Working Group. The Personal Health Working Group Final Report. Washington, DC: Connecting for Health: A Public-Private Collaborative, 2003.
3. Committee on Data Standards for Patient Safety, Board on Health Care Services. Key capabilities of an electronic health record system. Washington, DC: Institute of Medicine of the National Academies, 2003.
4. Mandl KD, Szolovits P, Kohane IS. Public standards and patients' control: how to keep electronic medical records accessible but private. BMJ. 2001;322:283–7.
5. Yasnoff WA, Humphreys BL, Overhage JM, et al. A consensus action agenda for achieving the national health information infrastructure. J Am Med Inform Assoc. 2004;11:332–8.
6. Kim MI, Johnson KB. Personal health records: evaluation of functionality and utility. J Am Med Inform Assoc. 2002;9:171–80.
7. Riva A, Mandl KD, Oh DH, et al. The personal internetworked notary and guardian. Int J Med Inf. 2001;62:27–40.
8. Huff SM, Rocha RA, McDonald CJ, et al. Development of the Logical Observation Identifier Names and Codes (LOINC) vocabulary. J Am Med Inform Assoc. 1998;5:276–92.
9. Scaleable Information Infrastructure Awards. National Library of Medicine. Available at: http://www.nlm.nih.gov/research/siiawards.html. Accessed Aug 24, 2004.
10. PatientSite. CareGroup. Available at: http://patientsite.bidmc.harvard.edu/. Accessed Aug 24, 2004.
11. Malamateniou F, Vassilacopoulos G. Developing a virtual patient record using XML and Web-based workflow technologies. Int J Med Inf. 2003;70:131–9.
12. Simons W. PING Talk Schema. Available at: http://ping.chip.org/schemas/ping-talk.xsd. Accessed Aug 24, 2004.
13. Simons W. PING Actions Schema. Available at: http://ping.chip.org/schemas/ping-actions.xsd. Accessed Aug 24, 2004.
14. Simons W. PING Talk Authentication Example. Available at: http://ping.chip.org/PingServer/exampleAuth.xml. Accessed Aug 24, 2004.
15. Rescorla E. HTTP Over TLS. Available at: http://www.ietf.org/rfc/rfc2818.txt. Accessed Aug 24, 2004.
16. Godik S, Moses T. eXtensible Access Control Markup Language (XACML) Version 1.0. OASIS. Available at: http://www.oasis-open.org/committees/xacml/repository/oasis-xacml-1.0.pdf. Accessed Aug 24, 2004.
17. Sun's XACML Implementation. Sun Microsystems. Available at: http://sunxacml.sourceforge.net/. Accessed Aug 24, 2004.
18. Sandhu RS, Coyne EJ, Feinstein HL, Youman CE. Role-based access control models. IEEE Comput. 1996;29:38–47.
19. Barkley J. Discretionary Access Control. Available at: http://csrc.nist.gov/publications/nistpubs/800-7/node25.html. Accessed Aug 24, 2004.
20. Barkley J. Mandatory Access Control. Available at: http://csrc.nist.gov/publications/nistpubs/800-7/node35.html. Accessed Aug 24, 2004.

21. Simons W. PING Credentials Schema. Available at: http://ping.chip.org/schemas/ping-credentials.xsd. Accessed Aug 24, 2004.
22. Simons W. PING Actor Attributes Schema. Available at: http://ping.chip.org/schemas/ping-actor-attributes.xsd. Accessed Aug 24, 2004.
23. Identity Systems and Liberty Specification Version 1.1 Interoperability. Liberty Alliance. Available at: http://www.projectliberty.org/resources/whitepapers/Liberty%20and%203rd%20Party%20Identity%20Systems%20White%20Paper.pdf. Accessed Aug 24, 2004.
24. Fowles JB, Kind AC, Craft C, Kind EA, Mandel JL, Adlis S. Patients' interest in reading their medical record: relation with clinical and sociodemographic characteristics and patients' approach to health care. Arch Intern Med. 2004;164:793–800.
25. Harmon A. As gene test menu grows, who gets to choose? New York Times. Jul 21, 2004;National Desk 1.
26. Wideroff L, Freedman AN, Olson L, et al. Physician use of genetic testing for cancer susceptibility: results of a national survey. Cancer Epidemiol Biomarkers Prev. 2003;12:295–303.
27. Sifri R, Myers R, Hyslop T, et al. Use of cancer susceptibility testing among primary care physicians. Clin Genet. 2003;64:355–60.
28. Ross SE, Lin CT. The effects of promoting patient access to medical records: a review. J Am Med Inform Assoc. 2003;10:129–38.
29. Szolovits P. The Guardian Angel Manifesto. Available at: http://www.ga.org/manifesto/GAtr.html. Accessed Jun 12, 2004.