Towards Large-Scale Sharing of Electronic Health Records of Cancer Patients

Praveen R. Rao
Computer Science Electrical Engineering
Univ. of Missouri-Kansas City,
Kansas City, MO, USA
raopr@umkc.edu

Stanley A. Edlavitch
School of Medicine
Univ. of Missouri-Kansas City,
Kansas City, MO, USA
edlavitchs@umkc.edu

Jeffrey L. Hackman
School of Medicine, Univ. of Missouri-Kansas City &
Truman Medical Center,
Kansas City, MO, USA
jeffrey.hackman@tmcmed.org

Timothy P. Hickman
School of Medicine
Univ. of Missouri-Kansas City,
Kansas City, MO, USA
hickmantp@umkc.edu

Douglas S. McNair
Cerner Corporation, Kansas City, MO, USA
dmcnair@cerner.com

Deepthi S. Rao
Veterans Affairs Medical Center, Kansas City, MO, USA
deepthi.rao@va.gov

ABSTRACT

The rising cost of healthcare is one of the major concerns faced by the nation. One way to lower healthcare costs and provide better quality care to patients is through the effective use of Information Technology (IT). Data sharing and collaboration and large-scale management of healthcare data have been identified as important IT challenges to advance the nation's healthcare system. In this paper, we present an overview of the software framework called CDN (Collaborative Data Network) that we are developing for large-scale sharing of electronic health records (EHR). In this on-going effort, we focus on sharing EHRs of cancer patients. Cancer is the second leading cause of deaths in the US. CDN is based on the synergistic combination of peer-to-peer technology and the extensible markup language XML and XQuery. We outline the key challenges that arise when sharing evolving, heterogeneous repositories and processing queries across multiple repositories. We present the novel architecture of CDN to overcome these challenges and discuss our plan for implementation, evaluation, and deployment.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical Sciences—*Medical Information Systems*

General Terms

Design

1. INTRODUCTION

The rising cost of healthcare is one of the major concerns faced by the nation. By 2019, it is projected that the nation would spend $4.48 trillion for healthcare. One way to lower healthcare costs and provide better quality care to patients is through effective use of Information Technology (IT). In a recent report by Stead and Lin [17], "data sharing and collaboration" and "large scale management of health care data" have been identified as key IT challenges to advance the nation's healthcare system. Vast amounts of information (e.g., electronic health records, drug data) remain largely untapped due to the lack of suitable IT solutions. Today, personal health information resides in digital silos and healthcare systems do not easily share information with each other. By tearing down these silos, vast amounts of clinical information can be utilized by medical practitioners and researchers to provide efficient, quality, timely, and cost-effective care to patients.

There are a growing number of local, state, and national level health information exchange (HIE) initiatives. The goal of HIE is to allow sharing of and access to clinical data to achieve Institute of Medicine's (IOM) vision of a learning healthcare system by providing safe, timely, effective, efficient, equitable and patient-centered care [4]. National Cancer Institute's caBIG is another nation-wide initiative whose vision is to speed up research discoveries and improve patient outcomes by connecting the members of the cancer community to share knowledge and data [7].

Achieving interoperability among applications processing clinical data has been a topic of interest for several years. Many advances have been made in developing standards for clinical data with regard to exchange/messaging, terminology, application, architecture, and so forth [10]. A federated database model, which is commonly adopted by today's data integration systems, allows a data provider (*e.g.*, clinic, hospital, research lab) to have full ownership and control over its data. Local access control policies can be implemented to protect the privacy of patients. But this model does not scale with increasing number of data sources and more complex schemas. This is because the process of creating a mediated schema and semantic mappings between the sources for processing queries becomes cumbersome and requires sufficient domain knowledge [17].

We are developing a new software framework called Col-

IHI'10, November 11–12, 2010, Arlington, Virginia, USA.
Copyright 2010 ACM 978-1-4503-0030-8/10/11 ...$10.00.

laborative Data Network (CDN) for large-scale sharing of EHRs. *Similar to caBIG,* CDN *aims to enable data sharing and query processing across multiple data sources.* CDN *differs from HIEs as it does not support electronic exchange of EHRs across participating health care providers.* In this on-going effort, we focus on sharing EHRs of cancer patients; cancer is the second most leading cause of deaths in the US. CDN leverages two successful technologies, namely, (a) peer-to-peer (P2P) computing and (b) and the widely adopted XML data model and XQuery language. The synergistic combination of these two technologies provides numerous benefits for sharing EHRs such as query expressiveness, ownership of data, minimal standardization, scalability and fault-tolerance. Using CDN, a single query can be issued by a user (*e.g.*, caregiver, medical researcher, patient) across multiple, heterogeneous data sources to perform aggregations and joins. Each data provider has complete control of its data and can employ local access control policies for protecting the privacy of patients.

In making CDN a success, we understand that there are several technical challenges to solve. The potential impact of CDN, however, can be huge. At the individual hospital level, it would allow more rapid review of events by quality improvement staff. It would significantly decrease the time spent gathering information from nationally reportable databases such as Core Measure and PQRI (Physicians Quality Reporting Initiative.) At the nation level, we can assess best practices, effectively monitor side-effects of new medicines, conduct comparative effectiveness research, and so on. Using CDN, superior decision support and data mining tools can be developed to operate on heterogeneous, evolving data sources.

The remainder of the paper is organized as follows. Section 2 provides the background and motivations. Section 3 describes the key challenges and the novel architecture of CDN to overcome these challenges. Section 4 discusses our plan for implementation, evaluation, and deployment.

2. BACKGROUND AND MOTIVATIONS

XML and Healthcare.

The extensible markup language XML has become the de facto standard for information representation and interchange on the Internet. It is widely adopted in domains such as service-oriented architectures and health informatics. XQuery is a popular query language for XML. It allows both the selection of qualifying nodes in an XML document and the creation of new elements and attributes and the specification of their contents and relationships.

In this work, we assume that clinical data is represented in XML using the widely adopted HL7 (Health Level 7) standards. HL7 is used by 90% of the hospitals in the US.[1] Organizations may have different requirements and produce HL7 messages conforming to their own XML schemas, which follow the HL7 Reference Information Model. (It specifies the grammar for these messages.) Of particular interest to us are the HL7 CDA (Clinical Document Architecture) and CCD (Continuity of Care Document) standards.

Distributed XQuery.

Distributed XQuery processing [14, 8, 20, 1] has been studied in recent years. The proposed solutions ship portions of a query to remote servers which then execute them. Locations of remote servers are explicitly specified in the query. None of these solutions support *location oblivious queries* that CDN aims to support. Essentially, a location oblivious query specifies only the data of interest but not where the data is located in a distributed environment.

With the growing popularity of P2P systems, many approaches have been proposed to find/locate relevant XML documents and their publishers in a P2P environment [9, 5, 6, 13]. Of particular interest is the *psi*X system that we have developed recently [13]. We show that *psi*X is well-suited to support location oblivious queries in CDN.

Federated Database Model and caGrid.

A federated database model is commonly adopted by data integration systems (*e.g.*, NCI caGrid [16], SHRINE [19], NeHii[2]). The design requires the creation of a mediated schema and semantic mappings of the sources for processing queries, which can be cumbersome as the local schemas become more complex and the number of data sources increases [17]. Furthermore, the mediation process requires sufficient domain knowledge of the data sources.

The underlying network infrastructure of caBIG, called caGrid, is a model-driven, service oriented architecture and the data services are accessed via grid services that expose data sources in a well-documented and interoperable form. The caGrid Federated Query Processor allows a user to perform distributed aggregations and joins over multiple data services using DCQL. Recently, caGrid was extended to support the creation of XML based data services. Thus, remote data sources can store data in native XML databases and can be accessed via caGrid services.

Motivations.

We describe two types of queries in caGrid that can be drastically improved by CDN in terms of functionality, performance, and scalability. The first type of query highlights the fact that currently caGrid does not leverage the full strength of XQuery effectively. The second query highlights that fine grained selection of data sources is desirable.

Consider a query (Figure 1) issued over Cancer data hosted by a network of participating research labs and medical institutions: *Find all the expression data where there are at least 50 conditions for genes found in the vacuole.* (This example is drawn from Summary and Initial Recommendations draft available on the website of caBIG.) The query performs joins across data exposed by three data services Gene, GeneOntology, and Microarray. Suppose we wish to access Gene and Microarray data from multiple providers. Then multiple queries should be posed – each one for a particular combination of Gene and Microarray data service – and therefore, will lead to poor scalability and performance when the number of data services grow. (Note that DCQL also suffers from a similar limitation.)

CDN overcomes this limitation by constructing a location oblivious query. Essentially, we use the collection() keyword in XQuery and replace service("http:// ... GeneService.wsdl") with collection("CDN") and service("http:// ... MicroarrayService.wsdl") with collection("CDN"), respectively. Now the query specifies a join over multiple data

[1] http://www.itl.nist.gov/div897/docs/Message_Maker.html

[2] http://www.nehii.org

```
FOR  $gene IN service
("http://cabio.osu.edu/GeneService.wsdl")/Gene,
$go IN service
("http://cabio.osu.edu/GeneOntologyService.wsdl")/GeneOntology,
$microarray IN service
("http://caarray.duke.edu/caArrayService.wsdl")/Microarray
LET $subject := $microarray/experiment/subject
WHERE
  $go/term='vacuole' AND $gene/goAcc=$go/acc AND
  $gene/gbAcc=$microarray/data/geneId AND
  count($microarray/data[geneId=$gene/$gbAcc]/condition)>50
RETURN
<subject>
  <subjectId>{ $subject/lsid }</subjectId>
  <species>{ $subject/species }</species>
  <microarrayData>
    { $microarray/data }
  </microarrayData>
</subject>
```

Figure 1: An example XQuery query

sources without specifying the locations of Gene and Microarray documents distributed across a network of participating data providers. The location service *psi*X will identify the data providers that contain relevant documents required for processing a location oblivious query. Consequently, a single query can be posed by a user, rather than a potentially large number of queries with location information.

Consider another query in caGrid that does aggregation over multiple data sources based on some selection criteria on the data (*e.g.*, gender = "male" AND smoker = "no"). Currently, the query will be shipped to each data source provided in the query. It is possible that not all specified data sources will have any matches for the selection criteria. A better approach would be if we send the query to only those data sources that contain matches. This is precisely what CDN can achieve – we can filter away irrelevant data providers based on the selection criteria in the query even without contacting them. We call this *fine-grained selection of data sources*. This is a big improvement especially in a wide-area network where bandwidth is a crucial resource and latencies are non-trivial. With fine-grained selection of data sources, queries can be processed efficiently.

3. THE PROPOSED ARCHITECTURE

In this section, we present the key design challenges arising in the design of CDN and the novel architecture of CDN to address these challenges. We are inspired by the success of P2P applications on the Internet such as Kazaa, BitTorrent, and Skype that are used by millions of users. We believe that the easier it is to publish/share data, the more is the incentive for data providers to participate. The complexity of the design should be hidden from users – the underlying framework should strive to achieve high performance, scalability, and high quality query results. We use the terms "data provider", "peer", and "participant" interchangeably.

3.1 Key Challenges

CDN focuses on sharing of EHR data of cancer patients, modeled as HL7 XML documents, across a large-number of participating data providers. Naturally, the data sources are heterogeneous and evolve with time. Our goal is to allow a data provider to participate with minimal standardization effort. Otherwise, it is a disincentive for the participation.

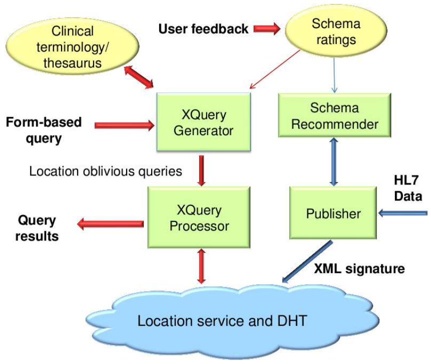

Figure 2: Proposed Architecture of CDN

We aim to answer the following questions: How can we share healthcare data on a very large-scale (e.g., petabytes of data)? How can we share and access heterogeneous data sources that evolve with time? How can we pose a single query to execute efficiently across multiple data sources? How can a data provider implement local access control policies like a federated system? Can we protect the privacy of patients and ensure security of patient records to ensure HIPAA compliance? Can we leverage clinical terminologies and thesauruses (*e.g.*, SNOMED CT [2], UMLS [3]) to identify similar concepts and terms?

We regard CDN's design goals to be similar to that of caBIG. CDN differs from HIEs and does not support the electronic exchange of EHRs to participating institutions. CDN is novel because of its query processing architecture: complex XQuery queries can be posed in a large-scale network and be processed efficiently. CDN leverages P2P technologies to achieve scalability and fault-tolerance. Minimal standardization is required to expose clinical data and CDN is responsible for generating appropriate queries to handle different terminologies used by data providers.

3.2 Key Software Components

The overall architecture of CDN is illustrated in Figure 2. The data provider are connected by an overlay network formed by the underlying Distributed Hash Table (DHT) [18, 15]. A DHT is a structured P2P network and allows new peers to join and leave the network at any time. Next, we describe the functionality of each component. While a few of the components have already been implemented and evaluated, the remaining are currently being developed.

Location Service used by CDN.

An important component of CDN is an Internet-scale location service called *psi*X [13, 12]. This service can be used to publish and locate/find XML documents of interest (using XPath (www.w3.org/TR/xpath/) in a P2P network. (See http://vortex.sce.umkc.edu/psix for a live prototype.) The *psi*X framework is built atop a DHT and inherits its properties such as scalability, fault-tolerance, and load balancing.

A user/peer can publish any valid XML document using *psi*X. (The document may or may not have a schema.) By publishing, we mean that an XML signature is generated for the document and stored in a distributed, hierarchical index. The XML signature essentially captures the summary of the XML document and includes both the structural summary and the value/content summary. The actual XML document is never stored in the network and resides

Type	Query
Incidence	What is the incidence of small cell lung cancer in a nonsmoker male between 2007 through 2010?
Staging	How many patients were diagnosed with prostate cancer stage II-B during 2005?
Regimen	What is the best treatment regimen for melanoma? What are the alternative regimens?
Radiation side-effects	What kind of cardiac side-effects were observed in patients receiving radiation to left breast?
Chemotherapy side-effects	How safe is R-CHOP regimen for my condition?
Survival rate	What is the 5-year survival rate for a patient with stage III-B colon cancer?

Table 1: Clinical Queries

with the publishing peer – the actual contents of the document are never exposed to other peers. When some peer issues an XPath query, *psi*X returns the names (or unique ids) of XML documents that contain a match for the query and their corresponding publishers. The actual documents can be requested by contacting the corresponding publishers. A publisher can deny access to its documents to any one at any time. Thus it controls its data (ownership) and can revoke access to its data at any time.

Joining and Leaving CDN.

An authorized data provider can join or leave CDN at any time and with little administration. While leaving, the data provider can delete the XML signatures from the distributed index. Unlike in typical P2P applications on the Internet, we expect low degree of churn in CDN.

Publishing an HL7 document in CDN.

A data provider is free to publish/share any valid HL7 XML document with other participants in CDN using *psi*X. The original HL7 XML document resides with the data provider. The provider has full ownership and control of its data and can implement local access control policies similar to a federated system – to protect the privacy of patients. The XML documents can only be modified by their respective owners.

A data provider can consult well-known schemas used by other providers. (Schemas are assumed to be listed, by choice, on a webpage that any participant of CDN can access.) Based on the specific needs, the data provider can either adopt or adapt a well-known schema by using a portion of the original schema and extending it with new elements and attributes. This works well when the provider has incomplete or missing data, or does not wish to expose certain attributes of the data due to privacy reasons, or has new attributes in the data. In our design, the advantage of modeling data based on well-known schemas is that CDN improves the likelihood of these documents being found during query processing. This is because of the way signatures are constructed by *psi*X. *This is a good incentive for a publisher to adopt well-known schemas.*

To track popular and unpopular schemas, we employ the idea of user ratings heavily used by Internet sites. A data provider can rate the schemas of others based on factors such as ease of adoption, ease of modification, etc.

The Schema Recommender automatically recommends appropriate schemas to a data provider to improve the query hits on the data published by the provider. It finds from the popular schemas those that are most similar to the schema(s) used by the provider. The provider is free to accept the suggestions or ignore them. The data provider can republish the data using recommended schemas.

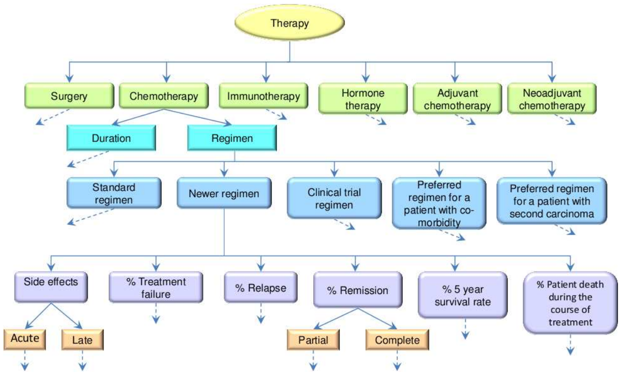

Figure 3: A Flowchart Fragment for Cancer Therapy

Posing Queries in CDN.

Our goal is to allow a user (*e.g.*, oncologist, patient) to ask general clinical questions pertaining to cancer diagnosis and treatment. A few examples are shown in Table 1. CDN will provide a form based interface for a user to construct queries. The form interface will be guided by clinical flowcharts. A flowchart fragment for cancer treatment is shown in Figure 3.

The XQuery Generator in CDN consults a clinical terminology/thesaurus to generate *location oblivious queries*. A user can specify a schema of interest or let the XQuery Generator pick from popular schemas. The query is then executed by the XQuery Processor, which is unique because it supports location oblivious queries by using *psi*X and new algorithms for efficient query processing in a large-scale network.

We highlight the main steps next. First, the query processor will extract the XPath expressions in the query to locate relevant XML documents in the network. Note that XPath expressions appearing in a query may navigate the same XML document. Rather than invoking the location service separately on each expression, we combine XPath expressions for the same document into a maximal XPath expression. The location service is then invoked on this maximal expression to identify the matching XML documents and their publishers/data providers. For example, in the query illustrated in Section 2, we can construct two maximal XPath expressions, one for Gene data and another for Microarray data, namely, `/Gene[goAcc][gbAcc]` and `/Microarray [experiment/subject][data/geneId]/condition`. The potential savings can be huge – both in terms of bandwidth consumption and network latency.

The next step is to bind the variables and expressions in the query (e.g., $gene, $gene/goAcc, $microarray/data) to nodes in the documents. Neither pure data shipping nor pure query shipping are the best choices in all scenarios in a distributed setting and a hybrid approach has shown to perform better [11].

Performance and scalability are important goals we aim to achieve in CDN. To realize these goals, we are developing a new abstraction called *Collection Tree* to gather partial query results from data providers that are relevant to a query. A *Collection Tree* is based on a dynamic programming algorithm that adapts to the number of relevant data providers for the query, the network locations of these data providers and other participants in the network, the amount of data that is being gathered, etc. This tree is rooted at the host where the query is initiated and the leaf nodes correspond to the data providers that contain matching documents for the query as identified by *psi*X and are willing to share the data. The actual documents are never sent across the network – only results of partial evaluation (*i.e.*, selections (σ) and projections (π)) are sent through the Collection Tree. Moreover, we encrypt the data flow through the Collection Tree. The internal nodes of the tree correspond to other willing data providers who wish to participate in the query execution. We are also designing new join algorithms using *Collection Trees*.

Security and Privacy.

CDN aims to provide high level of security to a participating data provider and protect the privacy of patient data. A data provider has complete control of its data and exposes only those that it wishes to share (*e.g.*, deidentified data of certain patients). Only authorized and mutually agreed upon data providers can join CDN. During query processing, the identify of the query initiator is verified by the data providers that contain matching documents. A data provider can refuse to execute a query if it believes that the query initiator does not have privileges to access its data. Data exchanged during query processing can be encrypted and therefore, prevents malicious attacks and eavesdropping. Note that never is an original document exchanged through the network and always resides at the publisher.

4. DISCUSSION

Our software development plan is committed to achieving the cornerstones of the caBIG initiative: open-access, open-development, open-source, and federation. We will modify an open source XQuery processor called SAXON (http://saxon.sourceforge.net) to process location oblivious queries. We will evaluate the effectiveness of CDN to handle multiple vocabularies in HL7 messages.

For sound evaluation of CDN, we will acquire good quality datasets, query sets, and schemas from Truman Medical Center and Cerner Corporation. We will extract, deidentify, and verify data with the help of the IT department and HIM coders. All through this process, we will require Privacy Board approval and IRB review. We plan to also utilize the deidentified datasets available from the ib2b project (https://www.i2b2.org).

We will conduct the evaluation on a local Gigabit cluster. We will deploy CDN within a couple of hospitals in Kansas City and conduct live studies and obtain feedback from caregivers. We will compare CDN with caBIG.

Acknowledgements.

Praveen Rao was partly supported by funds from University of Missouri Research Board.

5. REFERENCES

[1] DXQP - Distributed XQuery Processor. http://sig.biostr.washington.edu/projects/dxqp/.

[2] SNOMED Clinical Terms. http://www.nlm.nih.gov/research/umls/Snomed.

[3] Unified Medical Language System. http://www.nlm.nih.gov/research/umls/.

[4] Crossing the Quality Chasm: A New Health System for the 21st Century. *The National Academies Press, Washington D.C.*, 2005.

[5] S. Abiteboul, I. Manolescu, N. Polyzotis, N. Preda, and C. Sun. XML Processing in DHT Networks. In *Proc. of the 24th IEEE ICDE*, Cancun, Apr. 2008.

[6] E. Curtmola, A. Deutsch, D. Logothetis, K. K. Ramakrishnan, D. Srivastava, and K. Yocum. XTreeNet: democratic community search. In *Proc. of the 34st VLDB Conference*, pages 1448–1451, Auckland, 2008.

[7] D. Fenstermacher, C. Street, T. McSherry, V. Nayak, C. Overby, and M. Feldman. The Cancer Biomedical Informatics Grid (caBIG). In *Proceedings of IEEE Engineering in Medicine and Biology Society*, pages 743–746, Shanghai, China, 2005.

[8] M. Fernandez, T. Jim, K. Morton, N. Onose, and J. Simeon. DXQ: A Distributed XQuery Scripting Language. In *4th International Workshop on XQuery Implementation Experience and Perspectives*, 2007.

[9] L. Galanis, Y. Wang, S. R. Jeffery, and D. J. DeWitt. Locating Data Sources in Large Distributed Systems. In *Proc. of the 29th VLDB Conference*, Berlin, 2003.

[10] K. Kim. Clinical Data Standards in Health Care: Five Case Studies. http://www.chcf.org/∼/media/Files/PDF/C/ClinicalDataStandardsInHealthCare.pdf.

[11] D. Kossmann. The state of the art in distributed query processing. *ACM Comput. Surv.*, 32(4):422–469, 2000.

[12] P. Rao and B. Moon. An Internet-Scale Service for Publishing and Locating XML Documents. In *Proc. of the 25th IEEE Intl. Conference on Data Engineering*, pages 1459–1462, Shanghai, China, March 2009.

[13] P. Rao and B. Moon. Locating XML Documents in a Peer-to-Peer Network using Distributed Hash Tables. *IEEE Transactions on Knowledge and Data Engineering*, 21(12):1737–1752, December 2009.

[14] C. Re, J. Brinkley, K. Hinshaw, and D. Suciu. Distributed XQuery. In *Proc. of the Workshop on Information Integration on the Web*, pages 116–121, 2004.

[15] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In *Proc. of the IFIP/ACM Intl. Conference on Distributed Systems Platforms (Middleware 2001)*, Heidelberg, Germany, Nov. 2001.

[16] J. Saltz, S. Oster, S. Hastings, S. Langella, T. Kurc, W. Sanchez, M. Kher, A. Manisundaram, K. Shanbhag, and P. Covitz. caGrid: Design and Implementation of the Core Architecture of the Cancer Biomedical Informatics Grid . *Bioinformatics*, 22(15):1910–1916, 2006.

[17] W. W. Stead and H. S. Lin. Computational Technology for Effective Health Care: Immediate Steps and Strategic Directions. *The National Academies Press, Washington D.C.*, 2009.

[18] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *Proc. of the 2001 ACM-SIGCOMM Conference*, pages 149–160, San Diego.

[19] G. M. Weber, S. N. Murphy, A. J. McMurry, D. MacFadden, D. J. Nigrin, S. Churchill, and I. S. Kohane. The Shared Health Research Information Network (SHRINE): A Prototype Federated Query Tool for Clinical Data Repositories. *JAMIA*, 16(5):624–630, Sept. 2009.

[20] Y. Zhang and P. A. Boncz. XRPC: Interoperable and Efficient Distributed XQuery. In *Proc of Very Large Data Bases*, Vienna, Austria, September 2007.