Service-Oriented Architecture in Public Health

Interoperability Case Studies

Noam H. Arzt, PhD, FHIMSS

KEYWORDS

Public health, service-oriented architecture, registries, immunization.

ABSTRACT

Public health systems have been developed over many years and are costly to maintain or replace. Service-oriented architectures (SOA) have provided a way for these systems to remain viable and responsive to increasing demands for information and analysis. As healthcare entities look for strategies to effectively achieve "meaningful use:" of their EHR systems, SOA will emerge as one key technical strategy for enabling this functionality. This paper offers two case studies of core public health systems in different jurisdictions and the strategies used with SOA to extend system life and to enable new and important features.

Service-oriented architectures (SOA) have become quite popular among system architects and developers. Building upon such earlier concepts as application programming interfaces (API), and newer interest in system integration and interoperability, SOA is a powerful tool for conceiving of and delivering significant value to an organization through its IT systems. Because an SOA framework can be employed for both strategic (top-down) and tactical (bottom-up) implementations, it has been recognized as a powerful tool for application development and support.

Most major standards development organizations and standards harmonization efforts are now embracing SOA, and in many cases are trying to retrofit their approaches to accommodate this paradigm. In late 2008 Health Level 7 (HL7) and the Object Management Group (OMG) released a "Practical Guide for SOA in Health Care" under the auspices of their Healthcare Services Specification Project (HSSP) collaborative effort.[1]

In the Fall 2009 Integrating the Healthcare Enterprise (IHE) released a white paper titled "A Service Oriented Architecture (SOA) View of IHE Profiles."[2]

In the Fall of 2008 the Health Information Technology Standards Panel (HITSP) convened a special working group to explore and develop a service-oriented framework for its otherwise rigidly-conceived of specifications and constructs.[3] This work is continuing in the Spring and Summer of 2009 with the reframing of HITSP material to address key requirements of the American Recovery and Reinvestment Act of 2009.

This paper offers two case studies of core public health systems in different jurisdictions and the strategies used to use SOA to extend system life and to enable new and important features.

Fig. 1: System Evolution.

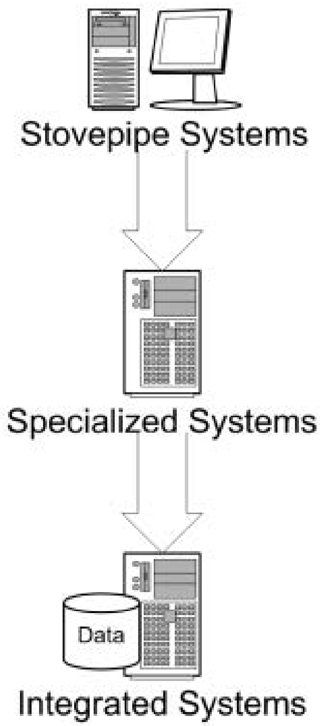


Table 1: CDC Applications.

CASA	Clinic Assessment Software Application (1992)
LITS	Laboratory Information Tracking System (1995)
PHLIS	Public Health Laboratory Information System (1989)
VACMAN	Vaccine Management System (1994)

PUBLIC HEALTH SYSTEM EVOLUTION

Over the past several years, public health systems have evolved significantly, both from a technical and programmatic standpoint. Many of these systems began in the 1980's (or even earlier) as program-specific, stove-pipe systems often based on aging mainframe or early, weak standalone personal computer technologies (Figure 1).

These systems originated in a variety of places. The Centers for Disease Control and Prevention (CDC) provided many such applications to health agencies thirsting for automation solutions. Users of these systems were often epidemiologists and others with public health analytical skills which they used to perform "programming" functions to tweak system functionality and performance to match the requirements of their jobs and to supplement the agencies' limited information technology staff. These agencies often had little or no internal capability or expertise to develop core information systems, as the information services function of these agencies was quite new and more focused on basic computing needs: desktop support, personal productivity applications like text processing and electronic spreadsheets, and basic network connectivity to support file and printer sharing. In these early years the Internet was just emerging as a mainstream, general purpose information highway; email was still a distant dream. Unfortunately, the CDC itself had limited funds and limited expertise in software development. What began as innovative applications over time became limited systems that did not keep up with changes in hardware and operating system capabilities, nor at times keep up with changing functional requirements (Table 1).

As personal computers became more powerful and operating systems became more useable with the advent of Microsoft Windows, two things began to occur. First, for a lucky subset of systems, CDC was able to update the products to make use of these more modern features and capabilities. Software was updated from DOS to MS-Windows. Additional printer support was added. In some cases, networking features were added to allow simple multi-user access. Second, public health agencies themselves began to recognize that information technology was a legitimate target for investment to improve their ability to perform core public health functions. Agencies began, on their own, to upgrade, replace, or create new systems that were more robust and specialized using modern database management systems and tools on more reliable platforms. The Internet began to come into its own, and the CDC promoted its first wide area communication and system integration projects through its Information Network for Public Health Officials (INPHO) initiative in 1993.

As the years went on, some agencies recognized the limitations of deploying systems purely within individual programs when the information related and their limited funds for technology could be better spent if leveraged across multiple projects. As applications became more network-aware and network-dependent, the need to leverage *network* investments became critical. Similarly, as systems moved to use more sophisticated relational database management systems (RDBMS) the pressure to share these expensive software licenses increased. These agencies developed a broader vision and some of their systems evolved into integrated systems supporting a wider variety of patient-centered or case-centered functions.

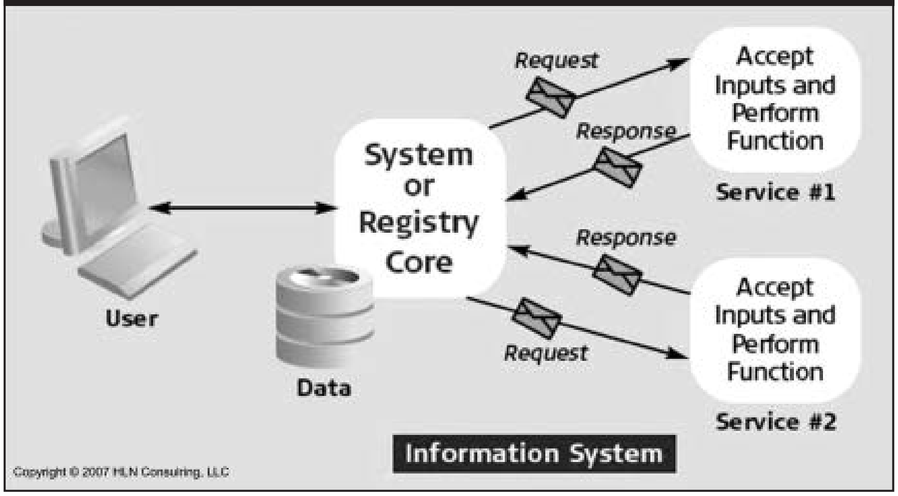
But funds have still been limited, and investments in information technology compete with other priorities within an agency. Extending the life of existing systems, while reducing their maintenance cost, is an important priority for any budget-conscience program.

PUBLIC HEALTH REGISTRIES

A public health registry is defined as, "...an organized system for the collection, storage, retrieval, analysis, and dissemination of information on individual persons who have either a particular disease, a condition (e.g., a risk factor) that predisposes to the occurrence of a health-related event, or prior exposure to substances (or circumstances) known or suspected to cause adverse health effects."[4] Immunization Registries, or Immunization Information Systems (IIS), are "confidential, computerized information systems contain vaccination histories and provide immediate access to a child's current immunization status to authorized providers. As a centralized, secure site of single record storage they provide an official immunization record for school, day care, and camp entry requirements."[5] Some of these systems have been in operation for ten or more years and continue to expand and evolve.

Early IIS were often deployed as client/server systems with traditional architectures.[6] As the Internet became more prevalent and stable, these client/server systems migrated to the World Wide Web either delivered through terminal services (e.g., Citrix) or rewritten as native Web applications. No single application technology currently dominates (e.g., .NET versus Java). These applications have provided two major functions: (1) provided an interface for data collection, as a primary purpose of the IIS is to consolidate immunization records from multiple sources; and (2) to provide decision support for the clinician to help ensure that patients (initially children, but increasingly adults as well) are immunized on time and, also, are not over-immunized.

Fig. 2: SOA Architecture.



While online applications began as a major tool for data collection, there has always been a strong movement within the IIS community to collect data electronically. The primary motivation for this strategy has been the existence of local provider-based information systems – both clinical systems and more administrative practice management systems – that are used by providers. Data is entered into these local systems and providers have been loath to re-enter data into an IIS application. Most IIS offer some capability for absorption of records extracted from these local systems, though these extracts are most often created in proprietary formats that vary from jurisdiction to jurisdiction. Efforts to standardize these formats have primarily focused on Health Level 7 (HL7) and more recently on IHE.

Interestingly, a small number of IIS have promoted the reverse model: providers are instructed to enter data into the IIS first and then exports are enabled from the IIS to the EHR-S or other local system. This strategy is aimed at ensuring that providers use the IIS for decision support, since most local systems do not have the complex algorithms necessary for assessing immunization history for a patient and accurately predicting whether the patient is up-to-date or in need of one or more immunizations. Certification criteria promoted by the Certification Commission for Health Information Technology (CCHIT) for both ambulatory and in-patient EHR-S do not yet include very many pediatric health functions, and Advisory Committee on Immunization Practices (ACIP) guidelines are some of the most complex to implement in systems.[7,8]

As more and more healthcare providers are using electronic health record systems (EHR-S) in their offices to maintain clinical records on their patients, IIS projects are developing the capability for interoperability with these EHR-S. This interoperability will become more important and necessary as providers seek to reduce the number of system interfaces and focus their data exchange on health information exchange networks (HIEN) that are emerging in their areas. Proprietary file formats for IIS data collection are being replaced by standards-based formats. But the push for standardization extends beyond that: providers are demanding standard data transport methods, as well as services-based capabilities to enable more seamless interoperability with their local systems.

SERVICE-ORIENTED ARCHITECTURES

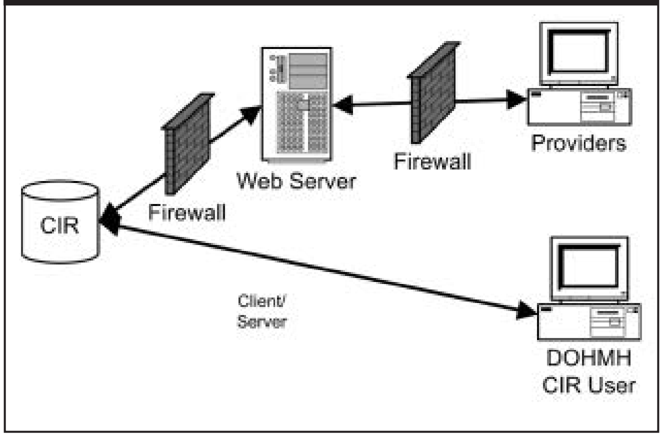
A wide set of material is available defining and describing Service-oriented Architecture (SOA) and the technologies commonly used to implement it.[9,10] Essentially, SOA is a building-block approach to application development which emphasizes re-use of software components that are built to perform individual functions and which interact with each other through clearly-defined interfaces.[11] For public health registries, SOA offers a powerful strategy to bridge old and new technologies, and it offers a way to enable interoperability with external systems that is increasingly required for the provider community with a minimum of retrofit and rework. SOA also offers a strategy for system enhancement through the acquisition or development of modular components that can change over time as needs change, as regulations change, and as clinical guidelines improve. A typical SOA architecture within a single system is displayed in Figure 2.

More extensive extrapolations of these basic concepts have been implemented in enterprise-wide deployments which, at their extreme, conceive of and deliver all system functionality as a set of services evoked as needed by participating systems.

SOA offers some distinct advantages for application development and support, including:

- Increased scalability through increased modularity: systems components can grow independent of each other as they are loosely coupled through services interfaces.
- Lower cost through software component reuse: no need to reinvent or redeploy a software module that can be reused in multiple systems.
- SOA is applicable either to entire systems or just to parts of systems, making it a flexible approach with no single "right answer" in the context of a particular application.

Fig. 3: CIR Architecture.



- SOA components tend to be more platform independent than other strategies, which is not to say that they may not be developed using platform-specific tools, but their interfaces are usually implemented in a platform-independent fashion. This allows services to be invoked by software objects that may not be developed or deployed on the same platform.
- SOA offers increased flexibility as services can be re-written and/or replaced as needs change with less impact on the overall system than other methods.
- SOA offers the potential for more agile and speedy system modification and improvement through its modular design.

There are some potential limitations, including:

- SOA implementations may run slower or require more processing power as data flows between loosely coupled components that may not be optimized for these data flows.
- There is a lot of hype in the marketplace over SOA, and it may be difficult to discern when components that are acquired are well-tested and operating properly.
- Just because a system is developed using SOA it does not mean it will be developed using good practices or appropriate methods. Just like any other form of software development, there can be "bad" SOA implementation.[12]

INTEROPERABILITY CASE STUDIES

Two case studies are being offered as examples of SOA implementations in public health. Both are implemented within IIS in different jurisdictions. In each case, there was a mature, operating IIS which required new functionality to be added to meet compelling business needs consistent with Centers for Disease Control and Prevention functional requirements for IIS[13] In both cases, an SOA approach was selected because it provided the best investment: a modular design which required the least modification to the existing, production system while retaining the greatest degree of flexibility for potential replacement or augmentation of the new functionality if requirements, standards, or technologies change.

NEW YORK CITY CITYWIDE IMMUNIZATION REGISTRY

First deployed in 1997, the New York City Citywide Immunization Registry (CIR) provides a consolidated database of all childhood and adolescent immunizations administered to City residents. Adult immunizations can also be stored in the CIR with consent of the patient. NYC's health code requires submission of immunizations for all patients age 18 and younger within fourteen days of administration [14]. As of June 2009, approximately 3.6 million patients are stored in the CIR along with over 40.5 million immunization events, making the CIR one of the largest IIS in the country. The database is initialized with records transferred from the New York City electronic birth system so all children born within New York City are automatically included (along with their initial Hepatitis B vaccination typically administered in the birthing facility). A simplified version of the CIR technical architecture is displayed in Figure 3.

Providers in the field access the CIR through a web-based application; internal DoHMH users access primarily through a client/server application with enhanced functionality. In 2004, the CIR was integrated with another City system called LeadQuest. Operated by the Lead Poisoning Prevention Program, LeadQuest (LQ) tracks the results from blood lead level tests required of all one and two year old children, as well as supporting the case management of lead poisoned children and the abatement of lead in buildings where it is found. This integration was achieved through the deployment of a central Master Client Index (MCI) in which patients from both the CIR and LQ are registered and de-duplicated. Figure 4 displays the architecture for this integrated system which went live in January 2004.

Over the years, a variety of mechanisms have been made available to providers to enable them to comply with this law, including on-line entry into a web-based application (now in its third incarnation), completion and submission of paper forms to an outsourced data entry service provider, and submission of electronic files extracted from clinical or billing systems for import into the CIR. In 2008, the CIR made a decision to eliminate the paper submission option. Initially, this option had been offered to allow a "low tech" strategy for those providers who could not or would not use either of the other two technology-assisted processes. Over time, as the proliferation of personal computers and the Internet even to the smallest practices continued, fewer and fewer sites were choosing this method of data submission. In addition, elimination of the outside service bureau processing the paper forms represented a significant cost savings to the agency.

At the same time, NYC's Department of Health and Mental Hygiene (DoHMH) was embarking on an ambitious project. The Primary Care Information Project (PCIP) "supports the adoption and use of Electronic Health Records (EHRs) among primary care providers in NYC's underserved communities. Its mission is to improve population health through appropriate technology and health information exchange."[15] A key strategy of this project is the acquisition and deployment of a commercial EHR-S for use among a set of smaller, safety net providers in the City. The software is being enhanced to support some key public health goals, including chronic disease management and infectious disease prevention. PCIP provides the software, initial training, and support for providers for whom at least 30 percent of their patient populations are either served by Medicaid or are uninsured. By June 2009 over 1,700 providers had signed up for this program; over 750 are already live.

Part of the vision for PCIP was strong integration of the EHR-S with the CIR. In this way, EHR-S users would be able to gain the

Fig. 4: Integrated Architecture.

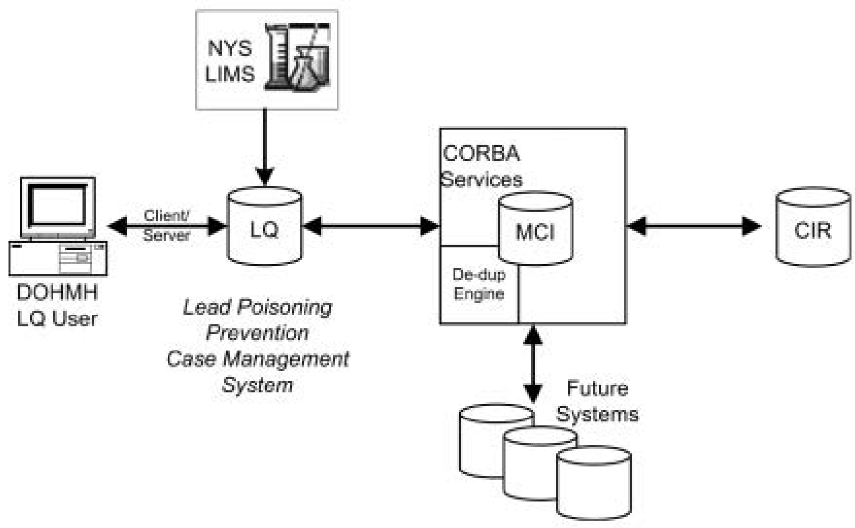


Table 2: Standards.

Transport	SOAP, using AXIS
Interoperability	Health Information Technology Standards Panel (HITSP) IS10 CDC Implementation Guide for Immunization Data Transaction using Version 2.3.1 of the HL7 Standard Protocol
Messaging	HL7 v2.3.1
Semantics	American Immunization Registry Association (AIRA) Immunization Information System Codebook

benefits of CIR data and special functionality without the need to leave EHR-S environment and log on to the CIR application. Specifically, NYC wanted the EHR-S to automatically:

- Report the patient's existing immunization history to the CIR.
- Obtain the patient's consolidated immunization history and clinical decision support from the CIR in real time.
- Add any missing immunization history to the patient's EHR and present the decision support to the provider at the time of treatment.
- Report any new immunizations administered by the provider to the CIR.

The volume of transactions expected for this interface was somewhat unknown, but it was important that the interface be able to scale as the demand scaled without impacting the rest of the CIR's operations. In addition, while PCIP was the initial target of this interface, other EHR-S vendors in the City were also interested in a seamless interface with the CIR, at least to facilitate their clients' required submission of immunization data. NYC's response to these challenges was the development of a set of web services, exposed to authorized partner systems over the Internet, to provide this functionality. The demand for real-time response to queries on the part of the CIR was an important factor in this architectural decision.

It was very important that, wherever possible, accepted national standards be used in this implementation. In this way, so long as participating EHR-S vendors complied with similar standards, interoperability might be assured. The standards that were employed are displayed in Table 2.

In addition, a NYC-specific Implementation Guide was developed and distributed to potential EHR-S partners with specific requirements for the data and interface based on the CDC Guide identified in Table 2 (it is common practice to develop implementation-specific guides, similar to HL7 or IHE profiles, to further constrain the general standards articulated in the master guide).

Two services were developed. The first service allows the EHR-S to query the CIR and receive in response the immunization history and clinical decision support for the patient. Specifically, the service provides these features:

- Accepts an incoming message after authenticating its source.
- Parses the standard HL7 VXQ message and retrieves patient identifying information.
- Performs deterministic search for the patient using available CIR features based on exact matches of key demographic fields.
- If necessary, performs probabilistic search for the patient using AI search engine through services made available by the MCI (see Figure 4) if deterministic match is inconclusive.
- Gets patient's immunization history from the CIR database and calculates the evaluations of those immunizations (i.e.., are they clinically valid or not) by invoking the standard CIR service available for this purpose, the Immunization Calculation Engine (ICE)
- Calculates patient's recommendations for all 13 routinely-administered vaccines (also via ICE)
- Constructs HL7 VXR response message
- The second service allows the EHR-S to report new immunizations to the CIR. Specifically, the service provides these features:
- Parses standard HL7 VXU message and retrieves patient identifying information from the CIR database
- Validates all of the demographic data
- Performs deterministic search for the patient using available CIR features based on exact matches of key demographic fields
- If necessary, performs probabilistic search for the patient using AI search engine through services made available by the MCI (see Figure 4) if deterministic match is inconclusive.
- If necessary, creates a new patient record in the CIR, otherwise updates patient demographic data in the CIR
- Validates all of the immunization data
- Inserts into the CIR any immunizations for which there is not already a record

Fig. 5: Web Services Architecture.

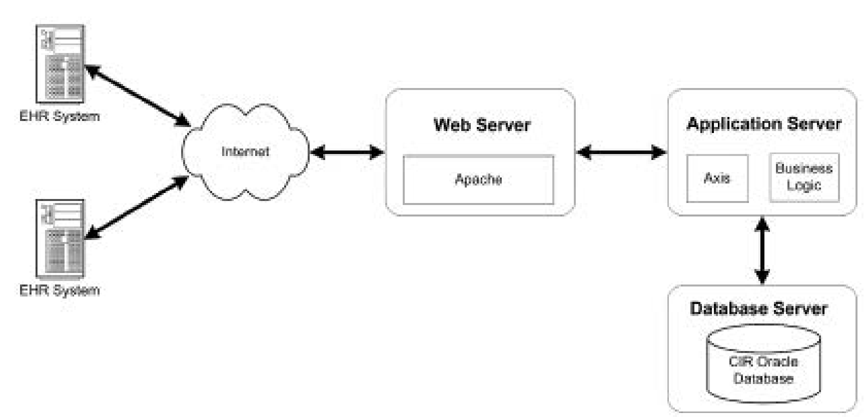


Table 3: Architecture Components.

Web Server	Apache
Application Server	Tomcat
SOAP Implementation	AXIS
Business Logic	Java & Stored Procedures
Database	Oracle

- Constructs HL7 acknowledgement or error message

Wherever possible, existing CIR features and functions were exploited rather than recreated. Additional business logic was added or expanded to accommodate new requirements. The architecture for these services is displayed in Figure 5.

Key components of the architecture are displayed in Table 3.

The CIR web services were developed in 2008 and have been tested by multiple EHR-S vendors. The services will be in production during the summer of 2009.

RHODE ISLAND KIDSNET

In 1997, the State of Rhode Island first implemented KIDSNET, a computerized registry used to track children's use of preventive health services. The registry brings together data from eleven different sources to assemble an overall picture, or profile, of the child's use of services. Programs include the Newborn Developmental Risk Screening, immunization, lead screening, hearing assessment, Women, Infant and Children (WIC), home visiting, early intervention, blood spot screening, birth defects, and vital records. Some of these programs use the centralized KIDSNET database as their system of record; others have external databases that routinely provide data to KIDSNET (and in some cases get data back in return). As of June 2009 there are over 285,000 children's records in the KIDSNET database.[16]

KIDSNET serves as the IIS for Rhode Island. Its central database serves as the IIS database, and the primary KIDSNET web application serves as the primary user application for providers in the field. This application began as a terminal-based application developed using Oracle Forms; only later was it replaced by a web-based application developed using Java. KIDSNET features changed and evolved over a number of years as requirements became clearer and funding for its continued enhancement became available. A simplified version of the KIDSNET architecture is found in Figure 6.

One of the core features of an IIS is the decision engine that evaluates whether a child's immunizations are valid according to accepted clinical practice, and calculates whether immunizations for the child are recommended (either for immediate administration or administration in the future). While ACIP guidelines for immunization evaluation are accepted nearly universally in the United States, the guidelines themselves are complex and allow for certain variations based on the vaccines used in a particular jurisdiction and some differences in clinical practice. Different immunization programs do in fact implement these guidelines somewhat differently. To that end, the implementations of these algorithms in IIS differ.

When KIDSNET was originally deployed it contained a rather simple algorithm for determining whether immunizations were valid and for predicting which immunizations were due for administration. As its use expanded, and the number of childhood vaccines steadily expanded, the agency determined that a more sophisticated algorithm was necessary. But the agency was unsure about choosing the best strategy for acquiring and integrating an algorithm into the existing KIDSNET product at a reasonable cost with as little disruption as possible to system operations. Several options were available. First, the agency could develop an algorithm from the ground up and add it into the KIDSNET code base as a new feature. This option would have taken a significant amount of time and funding, though it would yield a product that, if implemented correctly, would likely meet Rhode Island's precise requirements exactly.

Alternatively, Rhode Island could try to acquire an existing algorithm and somehow integrate it into KIDSNET. This option seemed conceptually feasible as the immunization algorithm typically accepts a limited number of inputs (the child's gender, date of birth, and immunization history) and produces a set of decisions. Even the identity of the child is not relevant to this activity. Several immunization algorithms were available – both commercially (for a fee) and from other IIS projects (typically made available to other public health agencies without fee). After studying and testing several alternatives, Rhode Island decided to integrate an immunization algorithm from a Western state that agreed to make it available without fee.

The algorithm being acquired was not developed as an SOA-enabled module, so there were in fact several options available for integrating it into KIDSNET. Though the agency could have chosen a traditional approach and inserted the new algorithm within the core KIDSNET code base, it decided instead that a service-oriented architecture would serve Rhode Island best for a number of reasons. First, as described above, the inputs

Fig. 6: KIDSNET Architecture.

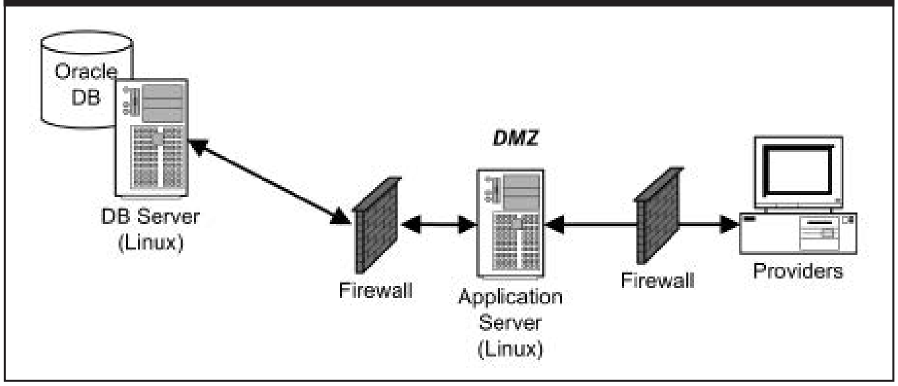
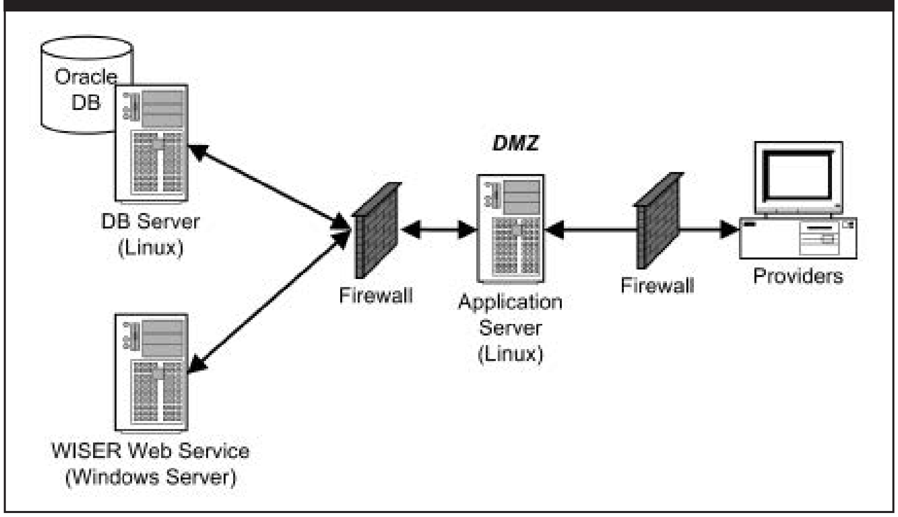


Fig. 7: KIDSNET SOA.



to the algorithm are known and straightforward, perfect for a parameterized interface. Second, while KIDSNET is an Oracle/Java-based system (see Figure 6), the algorithm being acquired was engineered as a Windows-based product written in Microsoft Visual Basic. Third, the process of integrating it directly into the KIDSNET code base would have been time-consuming and disruptive not only to KIDSNET operations but to other system development activities that needed to take place in parallel. Finally, implementing the algorithm in an SOA architecture would make it easier to replace the algorithm at some point in the future should the agency choose to do so.

So, KIDSNET was modified only slightly so that, when an assessment of a child's record was necessary, a web services call was made to the algorithm (called WISER, Web Immunization Service Evaluation and Recommendation) running on a separate server. KIDSNET sends the necessary data (child's gender, date of birth, and immunization history) and receives the assessment as an XML document. This data is then accepted by KIDSNET and stored in the database where it is available for the web-based application or ad hoc data query. The revised KIDSNET architecture is shown in Figure 7.

A more detailed diagram appears in Figure 8.

The client system, in this case KIDSNET, constructs the input XML loaded with the appropriate patient data. It then calls the Calculate web method and passes the input file. The web service then processes the data, creates the output XML file, and returns it to the client.

One additional benefit of this strategy became apparent. As more providers deploy EHR-S in their practices, they might want those systems to be able to evaluate a child's immunization history locally within the application instead of acquiring this data from KIDSNET. Through an SOA interface, it would be only a small incremental change for KIDSNET to allow external systems to address the WISER web service directly by sending it the necessary data and receiving a response. The revised SOA architecture is shown in Figure 9.

There are no standards for the structure or content of SOA-based implementations for an immunization algorithm, An effort in this direction was begun in 2007 by HL7 and OMG within HSSP, where work was done to develop an agreed-upon set of interface parameters and strategies for such an implementation (but not for the actual decisions within the algorithm which are not relevant to the interface), but the project was not completed nor has it advanced in several years.

CONCLUSION

Service-oriented architecture has moved into the mainstream of technical development strategies, and public health has not been left behind. Though not appropriate for every setting, this paper has demonstrated that it can be a useful tool to enable rich functionality within existing public health systems, while minimizing the cost and time to deploy these new features. SOA also provides some flexibility to replace system modules at some future date when superior modules become available.

In both NYC and RI, SOA was deployed in a limited fashion within the context of complex, mature systems that were already in place. In the case of NYC, SOA seemed like the most appropriate architecture for its HL7 interface which was has the potential to scale dramatically as EHR systems are deployed in the City. The web services deployment allows the services to change as national standards change with a minimum of impact on the rest of the system. In RI, the agency desired to acquire a software component that would have been expensive to purchase and difficult to develop on its own. Acquiring a functionally-acceptable module from another jurisdiction, and

Fig. 8: WISER Web Service.

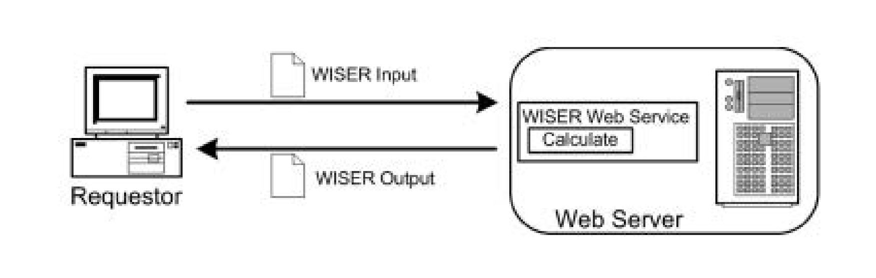
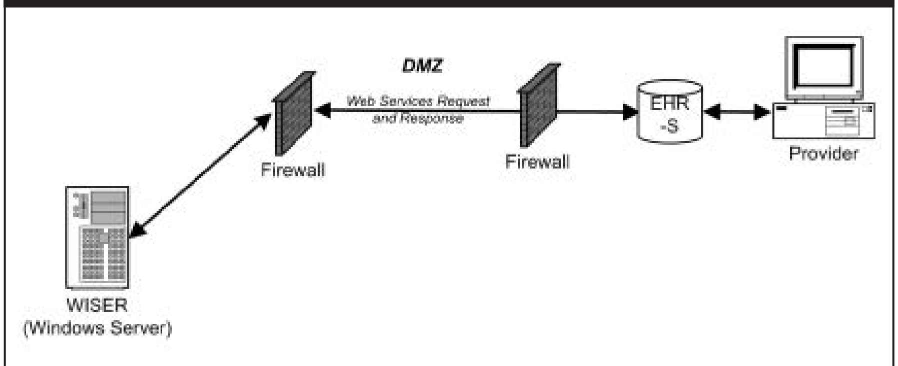


Fig. 9: EHR-S/KIDSNET Service Call.



implementing it in via SOA through a web services wrapper provided the best available solution that could be implemented fairly quickly, allow necessary customizations to be done, and offered an interface which could eventually be expanded to other system requestors with little to no additional effort.

Though the case studies offered in this paper come from public health, the experience and conclusions are applicable to other parts of the healthcare system as well. Hospitals and provider practices are also seeing an increasing need for interoperability with other systems outside of their organizations. SOA provides a useful strategy for either an incremental or more comprehensive approach to system redesign. Providers and hospitals will need to work quickly to achieve "meaningful use" of their EHR systems if they are to benefit from Federal incentives for their use (and later avoid penalties for non-use). Interoperability with public health systems in general, and with IIS in particular, is an explicit component of meaningful use which SOA can readily enable and support. **JHIM**

Noam H. Arzt, PhD, FHIMSS, is president and founder of HLN Consulting, LLC, San Diego, and does consulting in healthcare systems integration, especially in public health. He can be reached at arzt@hln.com.

REFERENCES

1. The Practical Guide for SOA in Healthcare: A Real-world Approach to Planning, Designing, and Deploying SOA. Healthcare Services Specification Project, Health Level Seven and the Object Management Group; 2008.

2. Painter J, Kirnak A, Moehrke J. A service-oriented architecture (SOA) view of IHE profiles. IHE IT Infrastructure White Paper. IHE International; 2009.

3. Healthcare Information Technology Standards Panel. www.hitsp.org.

4. National Committee on Vital and Health Statistics. FAQ. Available at: www.ncvhs.hhs.gov/9701138b.htm.

5. Every Child By Two. [Web site] Available at: www.ecbt.org/registries.

6. Arzt NH, Salkowitz SM. Technology Strategies for a State Immunization Information System," American Journal of Preventive Medicine, Supplement to 13(2), p. 52.

7. See http://www.cchit.org/

8. See http://www.cdc.gov/vaccines/recs/ACIP/

9. See http://www.omg.org/

10. See http://www.webservices.org/

11. The Practical Guide for SOA in Healthcare, p. 3.

12. Based in part on http://en.wikipedia.org/wiki/Service-oriented_architecture

13. See http://www.cdc.gov/vaccines/programs/iis/stds/min-funct-std-2001.htm

14. New York City Health code section 11.04 and 11.07(d) http://www.nyc.gov/html/doh/downloads/pdf/cir/healthcode2005.pdf

15. See http://www.nyc.gov/html/doh/html/pcip/pcip.shtml

16. See http://www.health.state.ri.us/family/kidsnet/