CHAPTER

8

Architectures and approaches to manage the evolving health information infrastructure

Brian E. Dixon[1,2], *David Broyles*[3], *Ryan Crichton*[4], *Paul G. Biondich*[2,5] and *Shaun J. Grannis*[2,5]

[1]Department of Epidemiology, Richard M. Fairbanks School of Public Health, Indiana University, Indianapolis, IN, USA [2]Center for Biomedical Informatics, Regenstrief Institute, Inc., Indianapolis, IN, USA [3]Marion County Public Health Department, Indianapolis, IN, USA [4]Jembi Health Systems, Cape Town, South Africa [5]Indiana University School of Medicine, Indianapolis, IN, USA

LEARNING OBJECTIVES

By the end of the chapter, the reader should be able to:

- Define the concept of a health information infrastructure.
- List and describe three key technical aspects of a health information infrastructure.
- Explain the concept of architecture from an information science perspective and how it is used in strategic planning for a health information infrastructure.
- Explain the role of transactions in health information exchange.
- List and describe the three types of interoperability important to health information exchange.
- Describe the role and core functions of an interoperability layer in supporting health information exchange.
- List and describe the components of an interoperability layer.

8.1 Introduction

This is the first chapter in the section of the book that covers the technical aspects of health

Health Information Exchange
DOI: https://doi.org/10.1016/B978-0-323-90802-3.00001-0

information exchange (HIE). In this chapter, we define and describe the key concept of a health information infrastructure, or the ecosystem in which HIE occurs. Information infrastructures are enabled by technical architectures, so we define a model HIE architecture and the core technical aspect of the model that facilitates HIE: the interoperability layer. Additional details on interoperability and the standards that support the syntactic form of interoperability can be found in Chapter 9. Semantic interoperability and standards are covered in Chapter 10. Additional components of a model HIE architecture are further detailed in Chapters 11−15.

8.2 The health information infrastructure

The health information infrastructure for a nation, state, or community is an ecosystem composed of the people, processes, procedures, tools, facilities, and technologies, which supports the capture, storage, management, exchange, and creation of data and information to support individual patient care and population health. While individual organizations such as hospitals possess discrete people (e.g., health care workers), processes (e.g., medication administration protocols, referral patterns), information systems (e.g., electronic health records, remote monitoring, laboratory information systems), and communities are composed of a larger set of technologies, organizations, and people that create a health information infrastructure [1]. Thus an information infrastructure is a sort of network of networks, and HIE is the method by which data and information are shared among the organizations, people, and technologies that comprise the defined ecosystem.

8.2.1 The health information infrastructure is an Ultra Large-Scale system

As described by the Institute of Medicine in its report on the digital infrastructure for supporting the Learning Health System [2], a health information infrastructure is similar to the concept of an Ultra Large-Scale (ULS) system as defined in computer science. An ULS system is a set of characteristics that tend to arise as a result of the scale of the system (in this case, the complex and fragmented organization as well as delivery of health care along with the heterogeneous nature of medicine and clinical data) rather than a prescriptive set of required technical components. Previous work on the ULS concept [3] has identified the following key characteristics of ULS systems:

- *Decentralization*: The scale of ULS systems means that they will necessarily be decentralized in a variety of ways—decentralized data, development, evolution, and operational control.
- *Inherently conflicting, unknowable, and diverse requirements*: ULS systems will be developed and used by a wide variety of stakeholders with unavoidably different, conflicting, complex, and changing needs.
- *Continuous evolution and deployment*: There will be an increasing need to integrate new capabilities into an ULS system while it is operating. New and different capabilities will be deployed, and unused capabilities will be dropped; the system will be evolving not in phases, but continuously.
- *Heterogeneous, inconsistent, and changing elements*: An ULS system will not be constructed from uniform parts: there will be some misfits, especially as the system is extended and repaired.
- *Erosion of the people/system boundary*: People will not just be users of an ULS system; they will be elements of the system, affecting its overall emergent behavior.
- *Normal failures*: Software and hardware failures will be the norm rather than the exception.
- *New paradigms for acquisition and policy*: The acquisition of an ULS system will be

simultaneous with the operation of the system and requires new methods for control.

8.3 Supporting the health information infrastructure

To organize and support an integrated health information infrastructure, where the various actors in the ecosystem can effectively exchange data and information, HIE networks must facilitate three important technical aspects of an infrastructure (the organizational, political, and legal aspects are considered in earlier chapters as well as the case studies):

- An *architecture* that defines how everything "fits together" and provides a common understanding for creating and managing the components of the infrastructure;
- Methods for *transactions* between information systems or the exchange of data and information; and
- Methods for *interoperability* or the ability for receiving information systems to do something with the data or information exchanged.

The specific information systems and applications used in HIE are defined and managed by individual hospitals, physician offices, health systems, public health agencies, and payors. These include systems such as laboratory information systems, radiology information systems, emergency department information systems, etc. In HIE speak, we often refer to them as point-of-care applications. Where an HIE infrastructure that can provide the most value is in defining the big picture and providing centralized technical services that enable data exchange among the various applications and information systems across the health care ecosystem.

8.3.1 Architecture

In computer science, the term *software architecture* refers to the high-level design of a system, the methods by which components are created, and the documentation of the system structures. In other words, the *architecture* of an information infrastructure defines how its various parts "fit together." Fig. 8.1 is an example of an architecture from the nationwide HIE in South Korea [4]. The image identifies the various components used by the HIE network, such as a registry server as well as a master patient index (MPI) server. In the context of South Korea's HIE network, the registry server contains information on clinical documents that exist for patients across various electronic health record (EHR)-based repositories. This is often referred to as a record locator service or RLS. As depicted in the architecture, the clinical document repositories are populated by EHR systems. Documents are then retrieved for viewing through either the DMZ (demilitarized zone) for clinics or the "Middleware" platform for hospitals. Architecture diagrams outline the various methods and interfaces needed to connect the various information systems and HIE network components together. The South Korea diagram further outlines the security measures used to keep data protected. An architecture like this one is derived from consensus discussions among the stakeholders involved in the network (e.g., health systems, public health authorities, pharmacies) who agree upon the components for exchange as well as the methods by which data and information will be exchanged.

When stakeholders agree upon an architecture, most often they agree to use one of several types of architectural patterns or styles. An *architectural pattern* is a general, reusable approach to a commonly occurring problem in software architecture within a given context like HIE. An *architectural style* is a reusable set of design decisions and constraints that are applied to an architecture to induce chosen desirable qualities [5]. While computer scientists disagree on the exact differences between the two idioms, they both provide a common

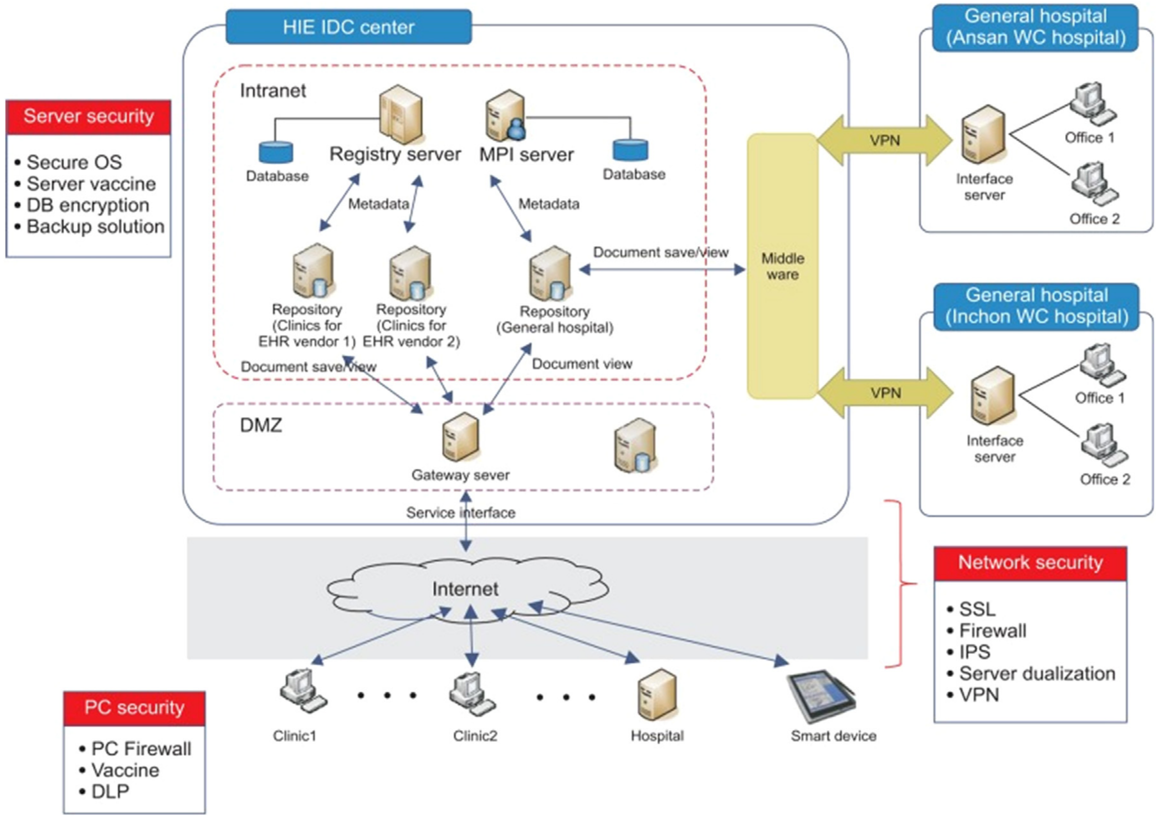

FIGURE 8.1 Example of an architecture from the nationwide HIE in South Korea. Source: *Originally published in Lee M, Heo E, Lim H, Lee JY, Weon S, Chae H, et al. Developing a common health information exchange platform to implement a nationwide health information network in South Korea. Healthc Inform Res 2015;21(1):21–9 and used with permission from the Korean Society of Medical Informatics.*

language or vocabulary with which to describe classes of systems [6]. Commonly used architectural patterns and styles include:

- Client/server—Distributed systems that involve a separate client and server system, and a connecting network. The simplest form of client/server system involves a server application that is accessed directly by multiple clients, referred to as a two-tier architectural style. Historically, the client/server style indicated a desktop application that communicated with a database server containing much of the business logic, or with a dedicated file server. This style is recommended when implementing business processes that will be used by people throughout an organization.
- Component-based architecture—Decomposes the system into reusable functional or logical components that expose well-defined communication interfaces. Essentially, the system is composed of prefabricated components that perform specific functions. A set of components function together as an application server. Aspects of a user interface are typically implemented as components. Other

common types of components are those that are resource intensive, not frequently accessed, and must be activated using the just-in-time (JIT) approach (common in distributed component scenarios); and queued components executed asynchronously using message queuing.

- Service-oriented architecture (SOA)—Application functionality is provided as a set of services that are loosely coupled because they use standards-based interfaces that can be invoked, published, and discovered. Services in SOA are focused on providing a schema and message-based interaction with an application through interfaces that are application scoped as opposed to components. SOA approaches are common for Software as a Service (SaaS) as well as cloud-based applications. The SOA style is recommended for message-based communication between segments of the application in a platform independent way, when you want to take advantage of federated services such as authentication, or you want to expose services that are discoverable through directories. Because services are independent and focus on a performing a narrow task, maintenance is easier when compared to other styles.
- Event-driven architecture (EDA) – In this paradigm, applications respond to events, which are significant changes in a state. For example, a patient is admitted to the hospital. The EDA is driven by *event notifications*, messages triggered by the event. The notifications communicate to other systems about the change in state with respect to a patient, procedure, provider, etc. Like SOA, the software components and services are loosely coupled. Systems that send notifications are referred to as *emitters* (event producers). Those that receive notifications are called *sinks* (event consumers). Sinks must react or respond to each notification, which might be to perform an action or filter, transform, and/or forward the event to another component.
- Microservices – A modern term that describes an architectural approach to a specific application rather than an enterprise (e.g., the HIE). SOA, EDA, and microservices all use a similar, loosely coupled approach and application programming interfaces (APIs), described below, to communicate between the various components within an application (e.g., medication hub) or HIE network.

Whereas client/server types are common for health information systems operating within a clinic, hospital, or health system, HIE applications typically use component-based or SOA types. Furthermore, SOA dictates that services operate in distributed environments and focus on document-centric communication [7], making this type very suitable for HIE since health information infrastructures share many aspects of an ULS system. More recent HIE initiatives have favored SOA approaches in their design and development, including the eHealth Exchange [8,9]. Some IT architects suggest that EDA and SOA models work well together, since EDAs are commonly built upon message-based frameworks and complement document-centric SOAs.

8.3.2 Transactions

A health information *transaction* can be defined as the exchange of electronic health information between two extraneous entities for a specific purpose. HIEs are designed to promote a variety of health information transactions. An example of a common transaction is the delivery of laboratory test results to the physician that ordered the test. The transaction occurs between the information system at the laboratory and the EHR system in the clinic or health system. Laboratory reports can also be

TABLE 8.1 Transaction classes commonly handled by health information infrastructures.

Transaction class	Description
Save/Update Records	Information regarding a patient encounter can be saved to the individual's electronic health record
	Demographic information about a patient can be updated in the CR
	A new patient is registered in the health system for the first time
Requests/Queries	A list of a patient's previous encounters restricted to a specific time frame can be retrieved
	Client information can be obtained by providing the client ID number
	Information about a specific health care facility can be retrieved if the point of service application calling the system provides the facility ID number
	A point of service system can request a list of a clients that match specific criteria outlined in the parameters of the query
	A point of service application can request a list of health care facilities that match specific criteria outlined in the parameters of the query
Alerts	The system can relay alert messages regarding patient care to physicians and other providers
	Public health authorities can be alerted to a new report of a disease like Ebola they may be closely monitoring in a state or nation
	Admission, discharge, and transfer messages are a special class of alert transactions commonly found in HIE environments

sent from the lab information system to other health system stakeholders such as a public health agency [10]. Thus transactions support routine business processes among health care system stakeholders. Other examples include sending medication orders to the pharmacy, retrieving a report from a specialty physician, and registering a new patient who visits the clinic for the first time.

In Table 8.1 we summarize common classes of transactions that can occur within a health system. While individual transaction types (e.g., medication order, lab result delivery) are designed around a specific process in health care, these transaction classes represent a broader set of transactions that can occur. There is a Save/Update Records class that represents the variety of transactions in which data or information about a patient is "pushed" or delivered from one system to another for either storage in the EHR or display to a clinician. Another class referred to as Requests/Queries represents transactions in which one information system asks or requests information from the EHR or another information system about a patient or population (e.g., all patients with diabetes, Veterans). A third class, Alerts, represents transactions in which a business rule is triggered by data in the system and a user is alerted about an emerging state (e.g., new blood pressure reading is high, diagnosis of a highly contagious disease).

8.3.3 Interoperability

Successful transactions between disparate information systems require multiple layers of interoperability including foundational, syntactic, and semantic [11,12]. Foundational interoperability refers to the technical infrastructure necessary to share information between systems [13]. Syntactic interoperability requires that messages sent between two information systems must be transmitted in a format that is recognized by both systems [14]. Semantic interoperability refers to the ability of one system to correctly decipher and process the information received from another information system without prior consultation [15].

8.3.3.1 *Foundational interoperability*

Disparate health information systems that wish to exchange information must adhere to a series of standardized communication protocols. Initially, the heterogeneous systems must be linked together to form a communication network. Many exchanges use Transmission Control Protocol/Internet (TCP/IP) connections to fulfill this requirement [11]. Information exchange also requires an application protocol such as the Hypertext Transfer Protocol (HTTP) [11,16]. Web services such as SOAP (Simple Object Access Protocol) or Representational State Transfer (REST) that define how and where to send the messages are also critical to structured information exchange [11]. Although foundational interoperability is necessary for data exchange, these requirements can be fulfilled relatively easily in relation to syntactic and semantic interoperability concerns.

8.3.3.2 *Syntactic interoperability*

Successful communication between systems also requires messages to be transmitted using a structure and syntax that is ascertainable to both systems [17]. Formats such as Extensible Markup Language (XML) are frequently used to satisfy this demand [11,14]. Enabling successful communication becomes increasingly complex as more heterogeneous systems with their own unique formatting are involved in the information exchange [11]. For this reason, HIEs have typically adopted messaging standards such as Health Level Seven (HL7) version 2 or 3 [4,11,18,19]. In Chapter 9, the book provides a more detailed discussion on syntactic interoperability, including HL7, and the process for developing and selecting technical standards.

8.3.3.3 *Semantic interoperability*

The ability to successfully receive a message does not assure that the system receiving the message will be able to interpret and complete a request. If external systems attempting to communicate with the HIE system use terminology or coding that is incompatible with the internal standards of the HIE, then the data must be translated into a standardized format before it can be interpreted. In addition, the response must be translated back into the language used by the external system. In many HIE structures, this can be a time consuming and expensive process [20–22]. In Chapter 10, the book more fully discusses syntactic interoperability as well as methods and tool for managing terminologies.

8.4 Open HIE—a model health information infrastructure

Interoperability between multiple disparate health systems presents a particularly complex challenge for HIEs to overcome [21,23,24]. The interoperability of health information systems is a prominent issue in the United States due to the fragmented makeup of the health information infrastructure [25]. Although maintaining many disassociated information systems empowers the United States to meet a diverse variety of local health care needs, this degree of flexibility also has some drawbacks [25]. Most health information systems were designed and structured to meet the unique needs of the implementing health care organization [26]. Planning for interoperability with external systems is generally a low priority in the development process [27]. Although standards for health data exist, many health care providers still identify clinical observations using locally specific terminology that is not harmonious with external systems [21]. As a result, health data are often collected in an inconsistent manner from one organization to another [28]. This often prevents the information systems of the separate organizations from corresponding with each other successfully without a uniform method of standardizing the data [28]. The lack of standardization that results from this approach makes it difficult to assemble the information from these fragmented local databases into a strong national system [25].

The *OpenHIE* framework is designed to promote the sharing of health information in countries with a diverse array of health information systems through a middle-out approach to national HIE [25]. The middle-out strategy adapts to the varying needs and capabilities of the heterogeneous entities involved in the HIE [25]. The OpenHIE framework facilitates interoperability by combining existing applications through relatively simple and inexpensive interfaces rather than implementing new systems or major re-development efforts [29]. This enables HIE efforts using the OpenHIE framework to avoid some of the financial and technical barriers experienced by many conventional HIE solutions [25,30]. As a result, OpenHIE can be implemented more quickly than other solutions and with few disruptions to any existing operational health information systems that will be participating in the HIE [18].

For numerous heterogeneous health information systems to communicate successfully, the architecture of an HIE must offer flexible processes and technologies that are capable of accommodating the constantly changing demands associated with health information. The OpenHIE model addresses this challenge through a component-based architecture. The component-based design enables multiple services to work together to provide a secure mechanism for sharing health care information while maintaining the flexibility of the system. This allows OpenHIE to meet the needs specific to the country where it is being implemented. As a result, OpenHIE has the potential to make the health care industry of an implementing nation more efficacious and boost the quality of service by enabling accurate and timely access to critical patient information [10,30–32].

The OpenHIE architecture is depicted in Fig. 8.2. Each component supports well-described, core health data management functions and interoperates with other components to ensure that health information from various point of service applications is rationalized to support person-centric and population-based health care needs. Reference implementations of each of the components exist to validate and highlight the functionality enabled within the architecture, and also are designed to support real world needs. Different compositions of these components can be used within a given environment to support myriad workflows.

The components of the OpenHIE architecture depicted in Fig. 8.2 include:

- *Interoperability layer (IL)* serves as the core method for connecting the components of the HIE with point of service applications. Detailed description of the IL is provided in the remainder of this chapter.

8.4.1 Business domain services

- *Shared health record (SHR)* that serves as a repository of person-centric records detailing visits, diagnoses, treatments, laboratory results, and other observations documented during care delivery or by public health authorities. Detailed description of the SHR is provided in Chapter 11.
- *Health management information system (HMIS)* that stores and redistributes population-level information normalized through the HIE. Detailed description of the HMIS is provided in Chapter 11.
- *Finance and insurance service (FIS)* that stores, categorizes, and facilitates the

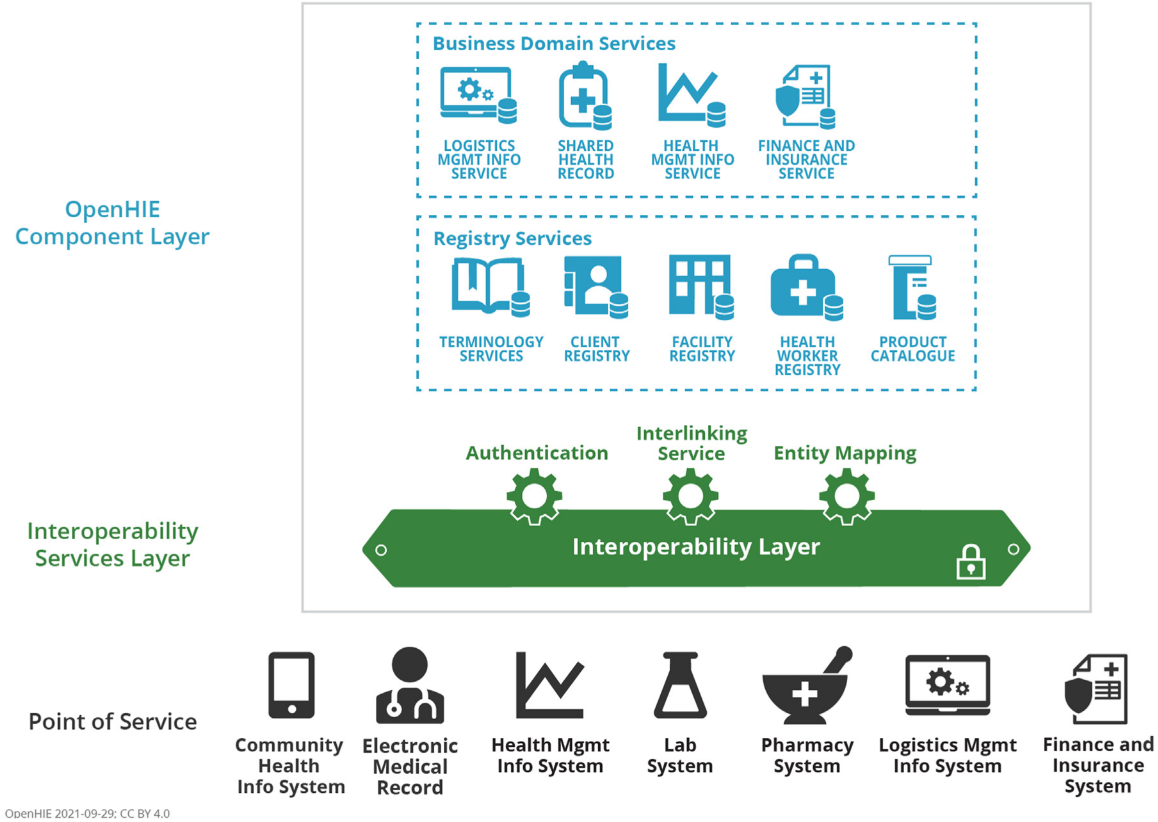

FIGURE 8.2 The OpenHIE framework that details architectural components, c.2021.

administration of centralized claims and finance-related data. The service receives claims/financial data and curates the management of them. Detailed description of the FIS is provided in Chapter 15.

8.4.2 Registry services

- *Terminology services (TS)* that manages local as well as reference terminologies, collections of unique concepts that describe diagnoses, treatments, and outcomes. Detailed description of the TS is provided in Chapter 10.
- *Client registry (CR)* that manages the unique identities of people receiving health services or who live in a community. Detailed description of the CR is provided in Chapter 12.
- *Facility registry (FR)* that manages unique health facilities located within a community, state, or nation. Detailed description of the FR is provided in Chapter 13.
- *Health worker registry (HWR)* that manages unique health workers at all levels of the health system, including doctors, nurses, pharmacists, social workers, and community health workers. Detailed description of the HWR is provided in Chapter 14.

Point of Service Applications are health information systems external to the HIE that have a need to interface with the various HIE components, which is why they are also depicted in Fig. 8.2. Mobile applications, laboratory information systems, health insurance claims systems, and home monitoring devices are examples of information systems that have a need to store data in the HIE and/or retrieve data from the HIE. These applications interact with the various HIE components via the IL.

8.4.3 Interoperability layer

In the OpenHIE model, the foundation of the flexible architecture is the *IL* [27]. The IL is a middleware system that enables easier interoperability between disparate information systems by bringing all of the infrastructure services and external applications together [33]. The OpenHIE structure includes both systems that request services and systems that provide those services. Service requestors are the external point of service applications such as a pharmacy information system or laboratory information system that are making requests of the internal OpenHIE components [27,34]. Service providers are the systems that accommodate these requests [27]. The service providers in the OpenHIE infrastructure consist of business domain services represented in Fig. 8.2 including the SHR and FIS as well as the registry services like CR, TS, and HWR. The job of the IL is to facilitate the transaction between the service requestor and the service provider.

The IL receives transaction requests from external client systems, conducts the correspondence between the internal components of the HIE, and possesses functions that can facilitate manageable interoperability between systems [35]. The IL serves several core functions that facilitate HIE for OpenHIE.

1. The IL simplifies the security for the HIE by providing a single point of access for all external systems that are attempting to communicate with the components of the HIE such as the CR, the provider registry, the FR, or the SHR.
2. The IL keeps a record of all of the transactions that take place in the OpenHIE.
3. The IL is responsible for routing messages to the appropriate service provider within the infrastructure.

In addition to the core functions provided by the IL, it also offers mediation functionality. If a point of service application sends a request in a format that is not recognizable to the service provider, the IL assists the transaction by furnishing an adapter that transforms the message into a format accepted by the internal components of the HIE. Similarly, the adapters will transform the data back into a format expected by the point of service application that initiated the transaction. When necessary, the IL orchestrates complicated transactions to remove that complexity from the consumer systems.

8.4.4 Structure of the interoperability layer

The IL was designed to address the problem of interoperability between health care systems through an architectural framework that incorporates the use of web services and progressive middleware [36]. The OpenHIE IL is composed of two components including the core component and the orchestrators and adapter services component. This architectural approach is based on the Enterprise Service Bus (ESB) model. An ESB is a middleware system that integrates messaging, web services, transformation, and routing to coordinate transactions among discordant applications [16,36]. These advanced middleware systems can be leveraged to overcome interoperability constraints between heterogeneous health information systems [16,33,37]. ESBs can accept transaction requests, and then perform mediation tasks to fulfill the transaction [16,27]. In addition, the

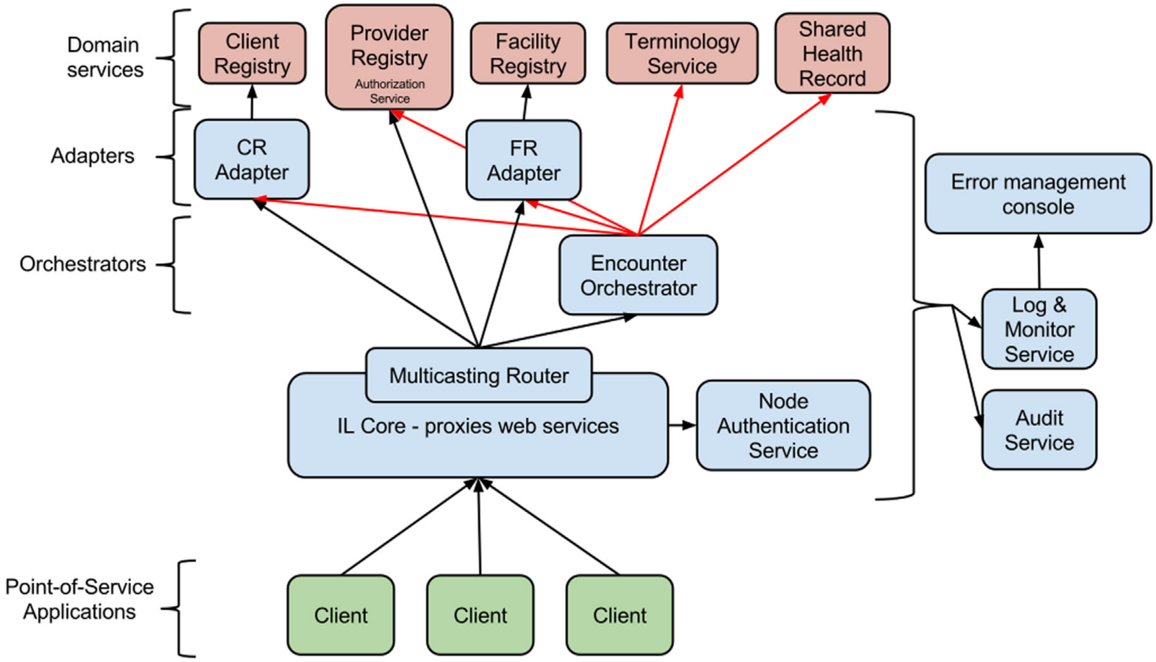

FIGURE 8.3 Architecture of the OpenHIE model illustrating the detailed components of the Interoperability Layer.

individual mechanisms of the ESB can operate independently of one another, which allows them to continue to function in the event that one component fails [27,38]. Fig. 8.3 illustrates the components of the IL and the role they serve in the OpenHIE infrastructure. The components are described in greater detail below.

8.4.5 Core component

The core component of the IL serves as the entry point into the HIE. It functions as a web service proxy and executes some supplementary tasks on the incoming requests to eliminate the need for other domain services to furnish them. It provides security for the HIE, logs transactions, identifies and displays errors that occur between services, and routes the incoming requests to the proper services. This requires an authentication and authorization service, a log service, and an audit service. When the IL receives confirmation from all of these services that the applicable information has been collected and saved, it passes the message to the router so it can be sent to the appropriate service.

Providing a single point of entry for all incoming messages serves to streamline the transaction process [39]. The IL can accommodate requests from a diverse range of systems through exposing an external PI [4,40]. The external API is publicly accessible to systems with the proper privileges and handles all transactions with point of service applications such as laboratory or pharmacy information systems. Requests from these external systems can be collected through an application protocol such as HTTP and then translated into a configuration that is recognizable to all components of the interoperability layer [27]. The use of an API enables OpenHIE services to be called in real time by external point of service systems. The point of service systems calling the IL can provide the necessary parameters for the system designated by the API, and then allow the IL to facilitate an automated response. This approach is advantageous over conventional systems that often relied on asynchronous approaches to interoperability [40].

8.4.5.1 Security

The core component of the IL also manages security for the entire HIE. Using the IL as the sole point of entry into the HIE mitigates potential security barriers to information exchange because the requesting systems can submit a message without ascertaining the location and security demands of the system that is providing the service [27]. The node authentication service depicted in Fig. 8.3 ensures that only authorized systems can interact with the components of the OpenHIE. Transactions that are received by the IL are acquired over a secured HTTPs connection through the external API. The IL uses the Audit Trail and Node Authentication (ATNA) profile from Integrating the Healthcare Enterprise (IHE) to authenticate transactions (see Chapter 9 for more on IHE). The ATNA profile relies on Mutual Transport Layer Security (MTLS) to ensure that transactions only take place when the digital certificate of a point-of-service application is trusted by the IL and the digital certificate of the IL is trusted by the point-of-service application.

The point of service applications that are calling the IL is responsible for authenticating users. For example, a provider may want to call the HIE to obtain a list of a patient's encounters within the last year. Once the provider has entered the correct password, they can submit a transaction request to the IL. This transaction request includes a provider ID number. When the IL receives an authenticated request, it calls the provider registry to verify the permissions associated with the provider ID number. This step verifies that the provider has the authority to make the request. This is important because some mobile point of service applications may be restricted to requesting data from specific services within the infrastructure, while other, more robust point of service applications may have unrestricted access to all of the services provided in the infrastructure. Once it is verified that the provider has the authority for the transaction, the IL will record the transaction in the SHR and notify the provider or other point of service application that the transaction was successful. If the provider does not have the authority for the transaction, an exception message would be returned to their point of service application. The results would be written to the audit log of the IL.

8.4.5.2 Monitoring/logging

Another function of the core component of the IL is to log and monitor the transactions that occur. The log, monitor, and audit services portrayed in Fig. 8.3 provide an audit trail for each transaction by storing every message as well as the crucial characteristics of the message including the sender's identity, details about the information that was received, the time and date the message was received, and the response to the message. This enables auditing when necessary, and provides insight into the movement of messages through HIE.

The logging mechanism of the IL also improves system performance and error-management within the OpenHIE infrastructure. Areas within the system that may be experiencing restrictions to the flow of information or inefficiencies can be identified more easily. All failed transactions are logged and can be grouped by the underlying source of the error. System administrators can use this information to correct recurring issues more efficiently [27]. Transactions that failed due to an internal system error can be re-run once the root cause of the error has been rectified without placing the burden of having to re-submit a request on the consumer.

8.4.6 Mediation component

The IL also offers a mediation component that executes the transactions that are routed from the core component. Mediators are microservices that carry out supplementary exercises on requests that are received by the IL [33]. These mediation microservices are disassociated components of the interoperability layer that each provides a distinct function. The two types of mediators that incorporated into the architecture of the IL include adapters and orchestrators. Adapters change the incoming requests into an acceptable format when necessary while orchestrators enable a business process to be carried out.

8.4.6.1 Adapters

Adapters allow the IL to transform incoming transactions into a format that the HIE recognizes [27]. Transformation ensures that the transaction will have syntactic interoperability. For example, a message that arrives in HL7 version 3 format may need to be converted into HL7 version 2 format in order to be recognized by the system because HL7 version 3 uses XML data objects to identify values while HL7 version 2 uses delimiters to separate values [19]. Adapters are also responsible for facilitating semantic interoperability for the transaction by translating the codes from the messages into the standardized language used by the IL. This requires a call to the terminology server that maps the standards used by the interoperability layer to the vocabulary used by the various participating systems [27]. The internal OpenHIE components such as the CR or FR each have a designated adapter as characterized in Fig. 8.3 to perform these services [35].

8.4.6.2 Orchestrators

The orchestration microservice enables the timely completion of transactions that require multiple tasks to be executed. The orchestrator may need to call numerous components of the HIE in order to assemble a valid response to the point of service application that initiated the transaction. For example, the completion of a request may require the verification of multiple identifiers that may be stored in separate registries within the HIE infrastructure such as the provider registry, FR, and CR. The illustration in Fig. 8.3 exemplifies how the encounter orchestrator may delegate tasks throughout the OpenHIE during a transaction.

These mediation services are summoned only when there is a requirement to translate or orchestrate a transaction. The core component will communicate directly with the domain service for transactions that do not require an adapter or orchestrator. The mediation services also log the messages that they dispatch to the domain services. The mediation components that are available in the IL are generally unique to the needs of the country that is implementing OpenHIE. Developing orchestrators and adapters as independent services within the architecture adds flexibility to the system because these subcomponents can be added or removed as needs change [35]. An alternative approach is to combine orchestration and adaptation services into a single component, referred to as an interface engine, which is described in more detail in Chapter 9.

8.5 Benefits of the OpenHIE interoperability layer

Collectively, the OpenHIE IL provides a solution to overcome many of the prominent barriers that can inhibit health information sharing. The ability to restrict access to a single entry point simplifies security, monitoring, and error management for the entire HIE infrastructure. The ability of the IL to process requests without the use of a specific message format is critical because no individual standards can exhaustively meet the demands of all prevailing and forthcoming information systems [28]. The flexibility of the IL structure enables OpenHIE to adapt as clinical guidelines, technology, and the needs of the stakeholders evolve over time [29]. Services can be added or changed as needed, and the loose component-based architecture can be leveraged to accommodate fluctuations in transaction volumes.

8.6 Emerging trends

Since the publication of the first edition, several important events triggered advancements in health information infrastructures across the globe. First, growth of the OpenHIE community amplified HIE efforts in multiple nations. Second, strong interest in nonclinical data gave rise to an evolving health information infrastructure that seeks to incorporate social, behavioral, and other data into HIE architectures. Finally, the COVID-19 pandemic triggered new investment in public health infrastructure, including HIE networks.

In 2016, the OpenHIE framework consisted of a few founding organizational partners and a handful of countries looking to implement HIE. Fast-forward to 2021, the OpenHIE community of practice now has over 10 organizational partners plus eight stories highlighting implementations of its architecture in various countries across Africa and Asia [41]. Even more impressive, the community has engaged health IT and HIE professionals from all continents working to evolve the framework and its components. Multiple nations involved in the OpenHIE community implemented standards and various architectural components during the past 5 years, and they incorporated the architecture into their nations' respective eHealth or digital health roadmaps. Their work amplifies the work of the global HIE community. The framework summarized in this chapter is an evolved form of the one first presented in the original chapter. Perhaps by the printing of the third edition, you might be part of the community and contributing to the next version of the framework.

Increasingly nations recognize the importance of the social determinants of health as well as other nonclinical factors that influence health and life expectancy. For example, access to clean water and nonpolluted air are some of the many nonclinical targets of the WHO, which seeks to promote health and well-being for citizens of all nations. Because HIE networks seek to support population health and improve health outcomes, many HIOs are working to incorporate nonclinical data into the health information infrastructure. This includes hosting data at geographic and population levels, which is quite different from the traditional patient-level focus of HIE. The trend toward supporting the capture, management, and use of nonclinical data is forcing many HIOs to re-examine their infrastructure and evolve their approaches to accommodate new data types and structures. This trend is covered in more detail in Chapter 19.

The COVID-19 pandemic exposed many flaws in the existing health information infrastructure [42]. Many public health agencies in the United States relied upon fax machines to receive laboratory confirmation of SARS-CoV-2 infections [43], and other data were captured using telephone calls from hospitals as well as manual data entry into web-based portals. Whereas HIE infrastructures were capturing patient-level characteristics and outcomes from hospitals and clinics, many public health agencies struggled to gather details on hospitalization rates as well as symptoms associated with the novel coronavirus. Given these challenges, the United States as well as other nations are investing in strategies to strengthen the health information infrastructure. For example, the American Rescue Plan Act of 2021 [44] appropriated over $50 billion to state, local, territorial, and tribal (SLTT) public health departments to, among other things, enhance their IT infrastructure and modernize their data systems. Other nations are contemplating their own digital public health strategies. All of these efforts are renewing interest in IT systems as well as HIE architectures that can help ensure during the next pandemic that patient and population health data are captured and shared electronically, and that public health authorities can readily use them to guide decision-making processes.

8.7 Summary

The technical attributes of a health information infrastructure are paramount to the effectiveness of large-scale information exchange efforts. HIEs must have a well-developed architecture, mechanisms for transmitting data from one system to another, and processes to provide interoperability between disparate information systems. In this chapter, we have discussed the core technical components of a robust health information infrastructure and described one model for achieving HIE within a community, state, or nation. While OpenHIE offers advantages, it is not the only way to achieve HIE. Regardless of the architecture selected, stakeholders who want to engage in HIE must make design decisions about the layers of interoperability or else the various, disparate information systems that exist within a given health care ecosystem will not be able to exchange data or information.

In the rest of this section of the book, we describe other important technical components of a health information infrastructure that can facilitate HIE. In Chapter 9, we discuss syntactic interoperability and available technical standards that support HIE between information systems. Then in Chapter 10 we describe semantic interoperability and available internationally recognized data standards for consistent representation of data and information across information systems. In Chapter 11 we discuss the role of a SHR that enables longitudinal examinations of individual patients as well as populations. Linking patient records from disparate facilities and health information systems requires a client or patient registry, which we discuss in Chapter 12. Then in Chapter 13 we describe facility registries, which uniquely identify the places where patients receive care in a health system. Next, we discuss health worker registries in Chapter 14 that unique identify the various individuals who provide care in a community or health system, including physicians, nurses, medical assistants, etc. Finally, in Chapter 15, we discuss the business domain service responsible for managing information about insurance and financial claims. When used together, these various components of a health information infrastructure not only facilitate HIE between information systems but they also enable providers, and community and public health organizations to answer questions about who receives care where in the community by whom as well as the outcomes of that care. This is ultimately the goal of HIE—to facilitate better quality, more efficient care that leads to improved health for individuals and populations.

Questions for discussion

1. What characteristics of the health system support the argument that a health information infrastructure can be classified as an ULS system?
2. Compare and contrast the HIE architectural styles and patterns. Which approach might be the most effective on a national scale?
3. Which type of interoperability is most important to architectural design of an HIE?

4. Why is a flexible HIE architecture so important in a country like the United States with multiple, heterogeneous health information systems?
5. What function provided by the interoperability layer is the most important to HIE endeavors? Why?

Acknowledgments

The authors thank the OpenHIE Community of Practice for sharing its collective knowledge and wisdom for use in the book, not only in this chapter but also in this entire technical section. The authors further acknowledge Mr. Ahmed Deeb, a research assistant from Earlham College, who contributed to the updates in the chapter for the second edition of the text.

References

[1] Dixon BE, Grannis SJ. Information infrastructure to support public health. In: Magnuson JA, Dixon BE, editors. Public health informatics and information systems. Health informatics. 3rd ed. Switzerland AG: Springer Nature; 2020. p. 85−104.

[2] Institute of Medicine. Digital infrastructure for the learning health system: The Foundation for Continuous Improvement in Health and Health Care: workshop series summary. Washington, DC: The National Academies Press; 2011.

[3] Feiler P, Sullivan K, Wallnau K, Gabriel R, Goodenough J, Linger R, et al. Ultra-large-scale systems: the software challenge of the future. Pittsburg: Software Engineering Institute, Carnegie Mellon University; 2006. Available from: http://resources.sei.cmu.edu/library/asset-view.cfm?assetID = 30519.

[4] Lee M, Heo E, Lim H, Lee JY, Weon S, Chae H, et al. Developing a common health information exchange platform to implement a nationwide health information network in South Korea. Healthc Inform Res 2015;21(1):21−9.

[5] Taylor RN, Redmiles DF, van der Hoek A. Architectural styles. Irvine, CA: Institute for Software Research, University of California. Available from: http://isr.uci.edu/architecture/styles.html.

[6] Microsoft Corporation. Architectural patterns and styles. Microsoft application architecture guide. 2nd ed. Redmond, WA: Microsoft Corporation; 2009.

[7] Koskela M, Rahikainen M, Wan T. Software development methods: SOA vs. CBD, OO and AOP. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi = 10.1.1.111.9582.

[8] Hosseini M, Ahmadi M, Dixon BE. A service oriented architecture approach to achieve interoperability between immunization information systems in Iran. AMIA Annu Symp Proc/AMIA Symp AMIA Symp 2014;2014:1797−805.

[9] Simonaitis L, Dixon BE, Belsito A, Miller T, Overhage JM. Building a production-ready infrastructure to enhance medication management: early lessons from the nationwide health information network. AMIA Annu Symp Proc 2009;2009:609−13.

[10] Dixon BE, McGowan JJ, Grannis SJ. Electronic laboratory data quality and the value of a health information exchange to support public health reporting processes. AMIA Annu Symp Proc 2011;2011:322−30.

[11] Gliklich RE, Dreyer NA, Leavy MB. Registries for evaluating patient outcomes: a user's guide [Internet]. 3rd ed. Rockville, MD: Agency for Healthcare Research and Quality; 2014.

[12] Broyles D, Dixon BE, Crichton R, Biondich P, Grannis SJ. The evolving health information infrastructure. In: Dixon BE, editor. Health information exchange: navigating and managing a network of health information systems. 1st ed. Waltham, MA: Academic Press; 2016. p. 107−22.

[13] Healthcare Information and Management Systems Society. HIMSS interoperability & HIE community. Chicago: HIMSS. Available from: https://www.himss.org/membership-participation/himss-interoperability-hie-community.

[14] Vergara-Niedermayr C, Wang F, Pan T, Kurc T, Saltz J. Semantically interoperable XML data. Int J Semantic Comput 2013;7(3):237−55.

[15] Dolin RH, Alschuler L. Approaching semantic interoperability in Health Level Seven. J Am Med Inf Assoc 2011;18(1):99−103.

[16] Andry F, Wan L. Health information exchange network interoperability through IHE transactions orchestration. International Conference on Health Informatics, Vilamoura, Portugal 2012.

[17] Healthcare Information and Management Systems Society. What is interoperability? 2013. Available from: http://www.himss.org/library/interoperability-standards/what-is-interoperability.

[18] Barbarito F, Pinciroli F, Mason J, Marceglia S, Mazzola L, Bonacina S. Implementing standards for the interoperability among healthcare providers in the public regionalized Healthcare Information System of the Lombardy Region. J Biomed Inform 2012;45(4):736−45.

[19] Viangteeravat T, Anyanwu MN, Nagisetty VR, Kuscu E, Sakauye ME, Wu D. Clinical data integration of distributed data sources using Health Level Seven (HL7) v3-RIM mapping. J Clin Bioinforma 2011;1:32.

[20] Lin MC, Vreeman DJ, McDonald CJ, Huff SM. A characterization of local LOINC mapping for laboratory tests in three large institutions. Methods Inf Med 2011;50(2):105–14.

[21] Dixon BE, Vreeman DJ, Grannis SJ. The long road to semantic interoperability in support of public health: experiences from two states. J Biomed Inf 2014;49:3–8.

[22] Baorto DM, Cimino JJ, Parvin CA, Kahn MG. Combining laboratory data sets from multiple institutions using the logical observation identifier names and codes (LOINC). Int J Med Inf 1998;51(1):29–37.

[23] Vest JR, Gamm LD. Health information exchange: persistent challenges and new strategies. J Am Med Inform Assoc 2010;17(3):288–94.

[24] Kuperman GJ. Health-information exchange: why are we doing it, and what are we doing? J Am Med Inform Assoc 2011;18(5):678–82.

[25] Coiera E. Building a National Health IT System from the middle out. J Am Med Inform Assoc 2009;16(3):271–3.

[26] Berges I, Bermudez J, Illarramendi A. Toward semantic interoperability of electronic health records. IEEE Trans Inf Technol Biomed: Publ IEEE Eng Med Biol Soc 2012;16(3):424–31.

[27] Crichton R, Moodley D, Pillay A, Gakuba R, Seebregts C. An architecture and reference implementation of an open health information mediator: enabling interoperability in the Rwandan health information exchange. In: Weber J, Perseil I, editors. Foundations of health information engineering and systems. Lecture notes in computer science 7789. Springer Berlin Heidelberg; 2013. p. 87–104.

[28] Gold MR, McLaughlin CG, Devers KJ, Berenson RA, Bovbjerg RR. Obtaining providers' 'buy-in' and establishing effective means of information exchange will be critical to HITECH's success. Health Affairs (Project Hope) 2012;31(3):514–26.

[29] Arzt NH. Service-oriented architecture in public health. J Healthc Inf Manage 2010;24(2).

[30] Adler-Milstein J, Bates DW, Jha AK. Operational health information exchanges show substantial growth, but long-term funding remains a concern. Health Aff (Proj Hope) 2013;32(8):1486–92.

[31] Dixon BE, Zafar A, Overhage JM. A framework for evaluating the costs, effort, and value of nationwide health information exchange. J Am Med Inf Assoc 2010;17(3):295–301.

[32] Vest JR. Health information exchange: national and international approaches. Adv Health Care Manage 2012;12:3–24.

[33] Crichton R. The open health information mediator: an architecture for enabling interoperability in low to middle income countries. Durban: University of KwaZulu-Natal; 2015.

[34] Liu S, Zhou B, Xie G, Mei J, Liu H, Liu C, et al. Beyond regional health information exchange in China: a practical and industrial-strength approach. AMIA Annu Symp Proc/AMIA Symp AMIA Symp 2011;2011:824–33.

[35] El Azami I, Cherkaoui Malki MO, Tahon C. Integrating hospital information systems in healthcare institutions: a mediation architecture. J Med Syst 2012;36(5):3123–34.

[36] Koufi V, Malamateniou F, Vassilacopoulos G. A big data-driven model for the optimization of healthcare processes. Stud Health Technol Inform 2015;210:697–701.

[37] Ryan A, Eklund P. The Health Service Bus: an architecture and case study in achieving interoperability in healthcare. Stud Health Technol Inform 2010;160(Pt 2):922–6.

[38] Van Den Bossche B, Van Hoecke S, Danneels C, Decruyenaere J, Dhoedt B, De Turck F. Design of a JAIN SLEE/ESB-based platform for routing medical data in the ICU. Comput Methods Prog Biomed 2008;91(3):265–77.

[39] Khan WA, Khattak AM, Hussain M, Amin MB, Afzal M, Nugent C, et al. An adaptive semantic based mediation system for data interoperability among Health Information Systems. J Med Syst 2014;38(8):28.

[40] Blobel B, Holena M. Comparing middleware concepts for advanced healthcare system architectures. Int J Med Inform 1997;46(2):69–85.

[41] OpenHIE. Our community is improving global health: OpenHIE; 2021. Available from: https://ohie.org/impact-stories/.

[42] Dixon BE, Caine VA, Halverson PK. Deficient response to COVID-19 makes the case for evolving the public health system. Am J Prev Med 2020;59(6):887–91.

[43] Kliff S, Sanger-Katz M. Bottleneck for U.S. coronavirus response: the fax machine. *The New York Times*. 2020 July 13, 2020;Sect. Upshot.

[44] American Rescue Plan Act of 2021, Public Law No: 117-2 (2021).