International Journal of User-Driven Healthcare
Volume 6 • Issue 1 • January-June 2016

An Architectural Solution for Health Information Exchange

Timoteus B. Ziminski, Department of Computer Science & Engineering, University of Connecticut, Storrs, CT, USA
Steven A. Demurjian, Department of Computer Science & Engineering, University of Connecticut, Storrs, CT, USA
Eugene Sanzi, Department of Computer Science & Engineering, University of Connecticut, Storrs, CT, USA
Mohammed Baihan, Department of Computer Science & Engineering, University of Connecticut, Storrs, CT, USA
Thomas Agresta, Department of Family Medicine, University of Connecticut Health Center, Farmington, CT, USA

ABSTRACT

Health information technology (HIT) systems including electronic health records (EHRs) have a market saturation nearing 92% at individual institutions but are still unsuited for cross-institutional collaboration of stakeholders (e.g., medical providers such as physicians, hospitals, clinics, labs, etc.) in support of health information exchange (HIE) of different HIT systems in geographically separate locations. In the computer science field, software architectures such as service-oriented architecture, grid computing, publish/subscribe paradigm, and data warehousing are well-established approaches for interoperation. However, the application of these software architectures to support HIE has not been significantly explored. To address this issue, this paper proposes an architectural solution for HIE that leverages established software architectural styles in conjunction with the emergent HL7 standard Fast Healthcare Interoperability Resources (FHIR). FHIR models healthcare data with XML or JSON schemas using a set of 93 resources to track a patient's clinical findings, problems, allergies, adverse events, history, suggested physician orders, care planning, etc. For each resource, a FHIR CRUD RESTful Application Program Interface (API) is defined to share data in a common format for each of the HITs that can then be easily accessible by mobile applications. This paper details an architectural solution for HIE using software architectural styles in conjunction with FHIR to allow HIT systems of stakeholders to be integrated to facilitate collaboration among medical providers. To demonstrate the feasibility and utility of HHIEA, a realistic regional healthcare scenario is introduced that illustrates the interactions of stakeholders across an integrated collection of HIT systems.

KEYWORDS

Cloud Computing, Data Integration, Data Warehouse, Fast Healthcare Interoperability Resources, Grid Computing, Health Information Exchange, Health Information Systems, Service-Oriented Architecture, Software Architecture, System Integration

INTRODUCTION

The healthcare domain, frequently criticized for its antiquated handling of data using paper-based patient registries in physician practices, has been infused with a multitude of software solutions for Health Information Exchange (HIE) that focus on the integration of patient data from multiple sources to improve quality of care, lower healthcare costs, and support research. Some important

DOI: 10.4018/IJUDH.2016010104

driving factors are the Meaningful Use Electronic Health Record (EHR) Incentive Program of Medicare and Medicaid (CMS, 2013) program to promote EHR usage or the Strategic Health IT Advanced Research Projects (SHARP, 2013) program. For example, SHARP has already spawned highly valuable platforms such as Informatics for Integrating Biology and the Bedside (i2b2, 2004) and the Substitutable Medical Apps & Reusable Technology platform (SMART, 2015). At the same time, patient involvement has been increased by initiatives such as Blue Button (Blue Button, 2013) which allows patients simple access to their data or the data of a cared for elderly parent or child collected from participating medical providers (e.g., physicians, nurses, clinics, hospitals, image labs, pharmacies, therapists, etc.). In addition, the fitness market has exploded with a variety of fitness devices (wearable technologies) that link to mobile applications with new initiatives by Apple and Google. Apple has proposed a new HealthKit app (Apple Health App, 2015) for a dashboard to manage health and fitness data, while Google has the Google Fit fitness tracker (Google Fit, 2015). Both companies are moving strongly into the smartwatch market to track motion, heart rate, blood pressure, activity, etc. In addition, Apple also announced ResearchKit (ResearchKit, 2015), an open source framework that allows researchers/developers to create apps in support of medical research; such a transformation will strongly rely on HIE to gather relevant data.

Despite the emphasis on HIE, there have been numerous problems that have been encountered during the same time span, particularly in regards to regional or statewide networks of connected healthcare stakeholders that practice HIE. Some very promising exchanges have failed (e.g., CalRHIO (Robinson, 2010) and CareSpark (Enrado, 2011)) and the progress of regional health information organizations has been described as “discouraging” and “insufficient” (President’s Council of Advisors on Science and Technology, 2010). Even though the adoption of health information technology systems (HITs) by medical providers is starting to approach a wider acceptance, the corresponding and required integration of healthcare data and systems via HIE remains a challenging problem, technologically as well as politically.

On the side of technology, factors that limit adoption of HITs and HIE have been identified (Gomes, Ziviani, Correa, Teixeira, & Moreira, 2012): a high development cost associated with HIE; a lack of agreed upon open-standardization particularly in regards to the sharing and exchange of data; a focus on brute force technology solutions rather than a healthcare process orientation that considers the needs of patients and providers and high data availability across HITs; and, a difficulty in maintaining HIE across multiple HITs that have the potential to evolve with new capabilities. On the political side, there is concern by major medical providers (e.g., hospitals and other healthcare providers in a particular region) that sharing data may lead to losing patients to competitors.

The Meaningful Use Stage 3 guidelines (HIMSS, 2015) were updated in 2015 and require all HIT systems to have cloud services to access, modify, and exchange health-related data, bringing particular focus to cloud interoperability (Baihan & Demurjian, 2017). HIT systems include electronic health records (EHR) and personal health records (PHR). In support of the interoperability and exchange of healthcare data, the international Health Level 7 (HL7) (HL7, 2016) organization has taken a leadership role for standards to allow the integration, sharing, and exchange of electronical healthcare data. Relevant standards incclude: HL7 Version 2 (HL7 V2, 20016), HL7 Version 3 (HL7 V3, 20016), the Clinical Document Architecture (CDA) (HL7 CDA, 2007), and, HL7 Fast Healthcare Interoperability Resources (HL7 FHIR) (HL7 FHIR, 2016). FHIR provides a RESTful Application Program Interface (API) to share data in a common format. The FHIR standard conceptualizes and abstracts information for HL7 into Resources that effectively decompose HL7 into logical components to track a patient’s clinical findings, problems, allergies, adverse events, history, suggested physician orders, care planning, etc. The intent is to allow a unified access to RESTful health-related data sharing APIs so that applications can be easily built to uniformly utilize multiple HIT systems. Concurrent with these activities has been an explosion of mobile health (mHealth) applications for both patients and medical providers (Aitken, 2013; UCSF, 2016). These mHealth applications also require access to health data via cloud services from multiple HIT systems to ensure that all of the necessary information is collected for patient care. Each of these HIT systems may operate with different paradigms (e.g., cloud, API, web services, etc.) and/or employ different security/access control techniques. Thus, mHealth applications need to work with a heterogeneous collection of paradigms and security protocols, with the strongly likelihood that set of information sources may grow or shrink over time.

The challenge is to define an architecture for HIE that is capable of supporting the ability of all of these existing HITs and applications to interact with one another utilizing the emerging FHIR standard in conjunction with various software architectural styles. Towards this challenge, this paper applies established software architectural alternatives and their best practices in conjunction with the emerging FHIR standards to propose a Hybrid HIE Architecture (HHIEA) (Ziminski, Demurjian, Sanzi, & Agresta, 2016) solution that integrates healthcare data from a set of HIT systems. The HHIEA solution is presented and explained in a five-step process by exploring the varied and complex requirements of the healthcare domain for supporting HIE of HITs, matching these requirements to established software architectural solutions that are then integrated with the concepts supported by FHIR. The first step overviews software architectural alternatives that can be chosen for structuring an HIE system that integrates multiple HITs. The second step describes a detailed and realistic regional healthcare scenario with multiple entities (i.e., medical providers and HIT systems) that defines the scope of stakeholders that includes: a sole-provider practice, a community practice, local and regional hospitals, testing laboratories (blood, scanning, etc.), pharmacies, a university academic medical center, etc.; this scenario was developed in collaboration with our co-author at a medical school. Within the scenario, the collaborative links between the involved entities are identified to serve as requirements of the domain. The third step examines the interplay and integration of the software architectural alternatives with the FHIR standard to demonstrate the cohesive way that they can interact for HIE of HITs. The fourth step proposes a hybrid HIE architecture (HHIEA) that leverages components from all of the identified alternatives combined with FHIR and highlights strengths for particular use cases. Finally, the last step maps the HHIEA into the assumed scenario, showcasing the way that the scenario satisfies the identified links and demonstrating the realization of the architecture in a system.

The remainder of this paper is organized into 6 sections. The Role of HIE section provides context information on varied HITs with a focus on the benefits of HIE and on the unique challenges that an HIE system architectures has to surmount. The Architectural Alternatives section explores high-level alternatives for system organization from the software engineering and architecture domains (federation, replication, and centralization) and a variety of instances of those alternatives, specifically: service-oriented architecture (Rosen, 2008), grid computing (Foster, 2002), publish/subscribe paradigm (Eugster, Felber, Guerraoui, & Kermarrec, 2003), data warehousing (Zeh, 2003), and cloud computing (Mell & Grance, 2011) and its strong link and interaction with FHIR. The Regional HIE Scenario section details a realistic regional HIE scenario of multiple HIT systems with a selection of identified stakeholders, their capacities as healthcare data providers and consumers, and the collaborative links that exist between them. The Hybrid HIE Architecture (HHIEA) section proposes an approach that leverages the studied architectural styles together with FHIR to address both informational and functional requirements of HIE for multiple HIT systems; in the process, the architecture is aligned to the realistic regional scenario. Then, the Future Trends section presents future efforts and directions in HIE that are emerging that may have an impact, including app-centric plugin architectures, including: the aforementioned SMART platform and its growing ecosystem for healthcare data-driven applications (Ziminski, De la Rosa Algarín, Saripalle, Demurjian, & Jackson, 2012), abstract architecture specifications for the construction of health applications such as the Open mHealth architecture (Open mHealth, 2015), the integration of genetic analysis and results into Electronic Health Records and Genomics (eMERGE, 2007), and, the evolution of some of these efforts with FHIR. Finally, the Conclusion and Ongoing Efforts section draws this paper to a close.

THE ROLE OF HIE

This section provides background material that is required for the remainder of the paper. To begin, health information exchange (HIE) concepts are introduced and placed into context that includes their usage with health information technology systems (HITs). Using this as a basis, the challenges for HIE architectures are identified and briefly reviewed. To complete the discussion, FHIR concepts are introduced and explained.

HIE and HITs

Medical and healthcare data obtained at the point of care presents both a challenge and an asset. Through transformation, aggregation, and analysis, healthcare data usage can include: billing and reimbursement through insurers to clinical decision support, automatic monitoring of patients and issuing of alerts, and enabling of research to establish new knowledge and improved procedures (Shortliffe & Cimino, 2006). To support all of these possibilities, HIE has emerged as one of the major means for electronic transfer of healthcare data among distinct healthcare organizations and their HIT systems. The goal of HIE is to make healthcare data available to healthcare stakeholders (e.g., healthcare providers, researchers in academia and industry, insurers, patients, etc.) in an efficient, cost-reducing, timely, and safe manner (Walker, Pan, Johnston, & Adler-Milstein, 2005). Monetary savings can be, for example, achieved by avoiding duplicated laboratory tests and the reduction of administrative overhead. HIE also has the potential to significantly improve the quality and safety of patient care by enabling healthcare providers to react faster to a patient's needs and avoid misdiagnosis, mistreatment, and adverse effects caused by incomplete knowledge about a patient's medical and health history.

Foundationally, HIE is based on the communication of a variety of complementary HITs employed by domain stakeholders. A core HIT are electronic health records (EHRs), which are a computerized record of a patient's health-related information (e.g., physician's observations, laboratory results, treatments, etc.) that can be created, managed, and consulted by authorized clinicians and staff within one or across multiple healthcare organizations. Their purpose can be manifold (support of research, education, etc.) but usually focuses on supporting continuity of care. There are various EHRs on the software market such as the publicly available Veterans Health Information Systems and Technology Architecture (VistA, 2003) developed by the US Department of Veterans Affairs, open source products such as OpenEMR (OpenEMR, 2012) and OpenMRS (OpenMRS, 2004), or a vast number of commercial products (ONC, 2015). A related HIT are personal health records (PHRs), which are electronic records of health-related information on an individual that can be drawn from multiple sources while being managed, shared, and controlled by the patient or their representatives. Thus, they will frequently contain a special subset of the data available in EHRs, but can also be a rich source of information such as the nutritional supplements or exercise regime and tracking of a patient. However, since their control lies with the patient, PHRs do not have the best reputation among physicians in terms of the reliability of the entered data. A prominent PHR is the Microsoft HealthVault (Microsoft HealthVault, 2007) platform with many insurance companies doing their own implementations.

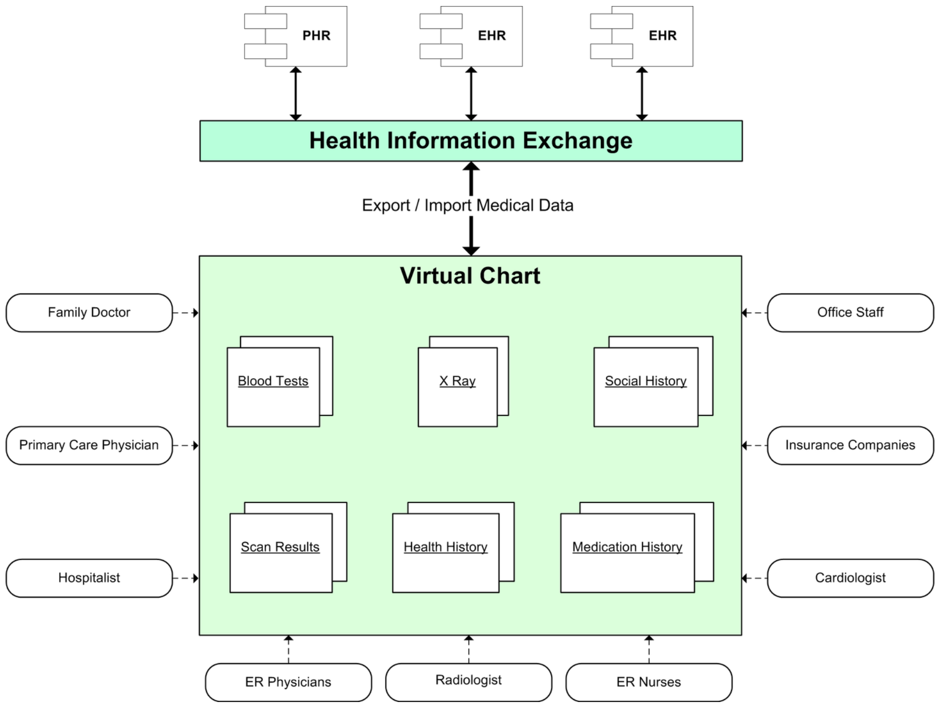
There are many other types of HITs. A Medical Laboratory Information Systems (MLIS) HIT for supporting the laboratory workflow from the test request to the specimen labeling to the creation of the lab report. Data Repositories or Data Warehouses HIT for structured information storage and support of research surveys in academia and industry. A Decision Support Systems (DSS) HIT for assistance with clinical decisions through evaluation of evidence-based knowledge in the context of patient specific data. Sample DSS functions include: drug interaction alerts or reminders for specific guideline-based interventions during healthcare, reminders for vaccine shots during a child's physical reminders to check blood pressure or glucose levels for individuals with chronic diseases, etc. A Practice Management Systems (PMS) HIT for processing financial, demographic, and nonhealthcare information about patients as well as scheduling, claim submission to payers, and other tasks.Electronic prescribing (e-prescribing) HITs for reviewing drug and formulary coverage and transmitting prescriptions electronically to a local pharmacy. E-prescribing can be integrated into EHRs and other HITs to also screen drug interactions and allergies. A billing system HIT supporting payers such as insurance companies or employers to process claims and bill patients.

Healthcare is a fundamentally collaborative discipline in which patients, particularly with chronic medical conditions, are treated by a wide range of medical providers in both outpatient (medical office or clinic) and inpatient (hospital) settings. This involvement poses a unique set of problems for data exchange including: a large number of stakeholders, workflows that routinely cross institutional boarders, and a lack of data/information standardization. Consider a case in which a patient who suffered from a heart attack presents himself with flu symptoms at a physician's office. When the physician attempts to retrieve an overview of the patient's medical history, their HIT might simultaneously obtain records from an EHR at the hospital's emergency room (ER), where the patient was provided with care, and an EHR at a rehabilitation clinic, where the patient went as part of the recovery process, and from the patient's insurance company, which covered the cost of treatment. While the record from the ER might be coding the patient's condition as Myocardial Infarction, the inpatient record can use the nonstandard description Heart Attack in combination with a decimal code, and the billing information might refer to a proprietary coding system that identifies the condition through an alphanumeric string. The significant challenge for HIE is to recognize that those three records belong to the same single individual and describe the same event (through a mapping of terms and semantics) to ensure that the medical history retrieved by the physician treating the patient's current flu symptoms lists the heart attack as one single incident, despite the incoherence and multiple locations of the source data.

This implies that to ensure correct and safe data transport between stakeholders, an HIE solution first has to overcome technical communication barriers between the participating systems, which can be only achieved by the establishment of nationally recognized standards for the transfer of healthcare data from both syntactic and semantic perspectives. In addition, an integrated HIE must not only enable the transport of the physical data, but also ensure that the meaning of exchanged information is maintained throughout the communication process. This second step can be achieved through a uniform utilization of medical ontology frameworks and common term databases which can translate heterogeneous names, codes, and identifiers to one common meaning (Demurjian, Saripalle, & Berhe, 2009).

Medical facilities which currently utilize HITs and would be involved in the above example are often not sufficiently interconnected due to a lack of HIE standards, competitive roadblocks to sharing data across enterprises, associated costs for HIE without proven cost benefits, etc. As a result, while data is instantly available inside the physician's office system, collaboration with external providers (and their HIT systems) is haphazard and ad-hoc. When a patient visits the majority of providers today, a hard copy paper patient chart is maintained, containing the medical records for all of the patient's visits over time. This chart grows over time as record deletion is extremely rare (unless incorrect data or lab results have been inserted); the time-oriented content (from present to past) is vital to clinicians who must not only treat the current ailment but also look back in time to have a full history of the patient and their conditions and treatments. On the path towards fully adopting HITs in all phases of a patient's treatment, the need to have a fully integrated, electronic version of this medication history (current and past) will be critical. As a result, the creation and availability of a virtual chart (VC) (Kenny, Parsons, Gratch, & Rizzo, 2008) has been proposed to provide a consistent, complete, and historically accurate patient medical record through automatic, HIE based integration of data from various source repositories. In Figure 1, the source repositories are a patient's PHR, a physician's EHR, and a hospital's EHR; other sources would include prescription records at pharmacies, billing records of insurers, imaging/scanning and laboratory test results, etc. In summary, the envisioned virtual chart: provides an individual patient EHR that gathers data from multiple HITs; allows for

Figure 1. Healthcare stakeholder collaboration via HIE and virtual chart. Source repositories (PHR and EHRs) contribute a variety of healthcare data such as blood test results to the virtual chart. The chart is then in turn used by healthcare stakeholders for collaboration and care provision.



retrieval of individual EHRs facilitating communication between providers; supports extraction of anonymized patient data from for aggregation to support data mining; has data monitoring capabilities for event tracking; implements security and access control enforcement and storage of audit trails that includes privacy rules for all parts of the stored data, data de-identification, etc.; issues alerts and preventive information that is in relation to a given chart instance; and, supports system personalization for adjusting to patient and provider needs.

Challenges for HIE Architectures

Software designers and engineers have worked on solutions for cross-institutional communication, (legacy-) system integration, and data exchange for decades and have created a large selection of solutions, templates, and alternatives that are routinely being leveraged by software architects in various fields on a daily basis. These well accepted methods meet a significant challenge in their usage for HIE due to the extremely complex and heterogeneous nature of the healthcare domain. In a typical scenario, each group of stakeholders utilizes different types of HITs; for each of these system types there are multiple products offered by different vendors and equipped with different communication mechanisms. Software architectures for HIE must support incremental development, since systems of this scale cannot be built in one effort. Similarly, HIE must support adaption to evolving requirements, since the list of requirements are not always fully known at the start of the system construction. Furthermore, scalability in throughput and size must be at the core of a suitable architecture, since processing will be increasingly challenging with HIE growth through the inclusion of more HIT systems, and both adoption and adaption of new technologies involved in the HIE. Finally, the chosen architectures for HIE must support heterogeneous environments, since the current HIT landscape is highly fragmented into many different types of HITs which in turn are available from different vendors. This typically hinders interoperability between systems and domain stakeholders.

Two types of interoperability are required by HIE. Syntactic interoperability enables interaction on a technical level (i.e., the format of the interaction) while semantic interoperability enables interaction on the content level (i.e., the meaning of the interaction). For example, connecting the HITs of a physician's office, hospital, and insurance company for simple data exchange could be achieved by establishing a message format that allows the exchange of events, containing a field "event name" (syntactic interoperability). However, the same event might be recorded differently, e.g., a heart attack represented as a cardiac arrest, a myocardial infarction, and a billing code, respectively in the involved HITs. The goal of semantic interoperability is then to recognize all three records as one event. To support this, a medical ontology would determine that a myocardial infarction is synonymous to heart attack. However, different HITs utilize different ontologies and may organize their ontologies differently, thus causing a need for a HIE architecture to be able to manage different ontologies and translate between them for semantic interoperability (Demurjian et al., 2009).

Integration of patient data requires an ID Management system with the ability to identify the patient across many disparate HITs. Since patients may see many providers and their information may be stored in different HITs, the HIE architecture must provide a method to uniquely identify a patient to retrieve their data form multiple systems. A lookup service allows an authorized HIT to query with a patient's demographic information to obtain a global identifier that other systems can recognize. Finally, HIE architectures also require a strong support for privacy and security, since they are tasked with processing highly sensitive data that is subject to regulations such as HIPAA (HIPAA, 1996) and FERPA (FERPA, 1974). Towards this goal, the architecture should implement a single point of entry to handle security, utilizing fine grained role and permission management for medical documents (De La Rosa Algarín, Demurjian, Berhe, & Pavlich-Mariscal, 2012; De La Rosa Algarín, Ziminski, Demurjian, Kuykendall, & Rivera Sánchez, 2013).

The Role of the FHIR Standard

The FHIR standard is primarily structured around the concept of FHIR resources (HL7 FHIR, 2016) which are the data elements and associated RESTful application programming interfaces (APIs) that can be leveraged for exchanging healthcare information, particularly between mobile applications and HIT systems. FHIR Resources hold the information that can be expressed in FHIR and are serialized XML or JSON format for exchanged from one health information technology system to another via RESTful API services. Resources are broadly classified into: Clinical Findings; Patient Problems, Allergies, and Adverse Events; Patient History; Suggested Physician Orders; and, Interdisciplinary Care Planning. To illustrate, sample FHIR resources from the 93 currently defined are: the Practitioner resource to track medical providers (physicians, nurses, office staff, etc.); the Patient resource can track demographic data on patients; the RelatedPerson resource to track parents/guardians; the FamilyMemberHistory for basic information on a family medical history; the Condition resource to track the relevant medical conditions; the Observations resource to track symptoms, and other medical observations; and, the Encounter/EpisodeOfCare resources to track the different times that changes to patient data occur based on a visit (Encounter) or action at the visit (EpisodeofCare).

One popular server for the FHIR standard is the HAPI-FHIR (University Health Network, 2016) open-source implementation written in Java. FHIR Resources can be utilized by HIT systems and mHealth applications for different purposes. For example, an mHealth application may use the Patient resource to store and exchange information about patients back and forth with different HIT systems. All FHIR resources have five main properties in common: a unique URL for identification purposes; common metadata; a human readable section; a number of predefined data elements; and, an extension element that enables a system to add new data elements. FHIR provides three equivalent representation formats: UML for a diagrammatic representation of the resource; XML that is subset of the HL7 schema for the resource; and, JSON to facilitate a programmatic exchange via a RESTful

API. FHIR supports a number of REST API services to enable a system to retrieve and modify data in the Resources. The main five services are: Create to add a new instance of a resource; Read to retrieve an existing instance of a resource; Update to manipulate data in an existing instance of a resource; Delete to remove an existing instance of a resource; and, Search to retrieve all existing instances of a resource.

ARCHITECTURAL ALTERNATIVES

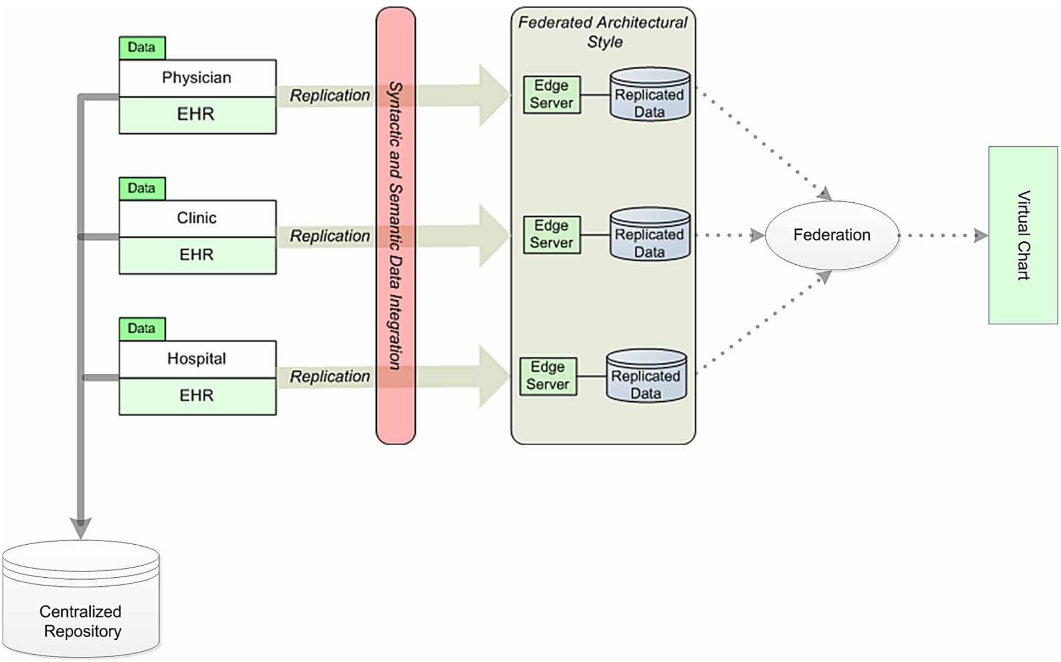
To set the context for the proposal of a hybrid HIE architecture, this section presents software architectural styles that have the potential for usage in data integration and HIE including the emergent FHIR standard. The first three are collectively reviewed, namely: federation that directly and in real time utilizes the data of HITs that is available for providing care; replication that off-loads data from HITs to dedicated edge servers that are then in turn federated; and, centralization that extracts data from multiple HITs into a single shared repository. Using this as a basis, a set of well-proven architectural styles and their suitability for HIE are analyzed, namely: service-oriented architecture (SOA), grid computing, publish/subscribe architecture, and data warehouse. Finally, cloud computing and its strong relationship to the FHIR standard for both interoperability and the ease of creating new mobile and desktop applications, are examined.

Federation vs. Replication vs. Centralization

In the federated architectural style, shown in the right side of Figure 2, healthcare data remains at its HIT source(s), and is made remotely accessible on demand. In this classic approach, a global database query is submitted, broken down into constituent local queries, processed at the remote sources, with results collected and formatted for presentation to the user. There are many advantages to this style. First, since healthcare data remains under the governance of the HIT source(s), the central portions of such architectures can remain relatively lightweight and process ad-hoc data queries on demand. As a result, federated solutions lower the amount of sensitive data that is shared and avoid making a priori assumptions about the value of data that an exchange participant may offer. Second, the data that is provided by a federated solution should be the most up-to-date version available since the federated user is accessing the federation in real-time. Third, the nature of federated solutions promotes scalability in terms of adding new HIT sources to an exchange network. There are also some significant disadvantages. First, the availability of data depends on the availability of the providing HIT source(s) and may therefore be unpredictable, e.g., patient records stored in a physician's office EHR may be unavailable after office hours. Second, performance bottlenecks are possible, e.g., the major hospital's EHR in a region may be accessed from numerous physician offices on a daily basis; if such access impacts performance to delay patient care, the result can be catastrophic. Third, security and monitoring of activities may be challenging in federated solutions, especially controlling and restricting access to sensitive data required distributed security models.

To overcome some of the fundamental issues of the federated approach, it can be extended by replication, as shown in the middle portion of Figure 2, which introduces an additional repository for each HIT source housed on an edge server. Since each repository is periodically updated, one advantage is that there is no impact on patient care on the HIT source. A second advantage is that the data integration to the edge servers gives each participant fine-grained control about which data is shared in an exchange, therefore preserving local governance choices. Third, the replicated repositories can be optimized towards performance and availability, with minimal impact (typically only during offloading time) on HITs that are actively used for providing care. However, there are some disadvantages. First, the data that is offloaded from HIT source(s) may not always be up-to-date; periodic updates may be limited to minimize impact on the usage of the HIT source(s); for patient care, this is a problem since medical providers always require the latest patient data. Second, the information may be out of sync, e.g., the same patient that has data in two or more different HIT(s)

Figure 2. The federated, replication, and centralized architectural styles. Healthcare data is collected and used in EHRs of different stakeholders. Replication decouples the data from the sources via a transfer to designated edge systems. Federation allows a near real-time combined view on the replicas through the virtual chart. Centralization in a separate repository provides an alternative view on the data after transformation and aggregation.



may have an inconsistent view if the update periods differ per source. Third, security and monitoring remain just as challenging as with a federated approach.

Lastly, the centralized approach in the bottom left of Figure 2 provides a common location—a new central participant in the data exchange—to operate as a main, shared repository. Designing and implementing a centralized architecture requires the extraction and integration of existing data from the HIT source(s), either stepwise or in one major effort; this is true for both initialization and periodic updates to the repository's content. Through this integration process, a centralized approach makes it possible to attain syntactic and semantic interoperability, as discussed in the prior section. A first advantage is that single administrative governance can be utilized to control access to the shared information. Second, since the data in the central repository is available independent of HIT source(s), these source(s) are no longer impacted by external access. Third, contributors have significant control in terms of the patient data that is to be shared and in what way (security). However, there are some disadvantages. First, there is a capacity factor to be considered since the data from potential hundreds of providers (e.g., hospitals, clinics, practices, labs, etc.) in a geographical region must be collected and combined; a country-wide integration would be even more difficult to achieve in practice. Second, the integration process from multiple HIT source(s) would require the need to reconcile all of the patient data to insure that the same John Smith's data has been collected from all sources without error. Third, from a system perspective, there is a probable performance bottleneck as HIT source(s) increase, the potential for fatal events if the repository goes down, and attacks or data theft that now impact a larger body of data.

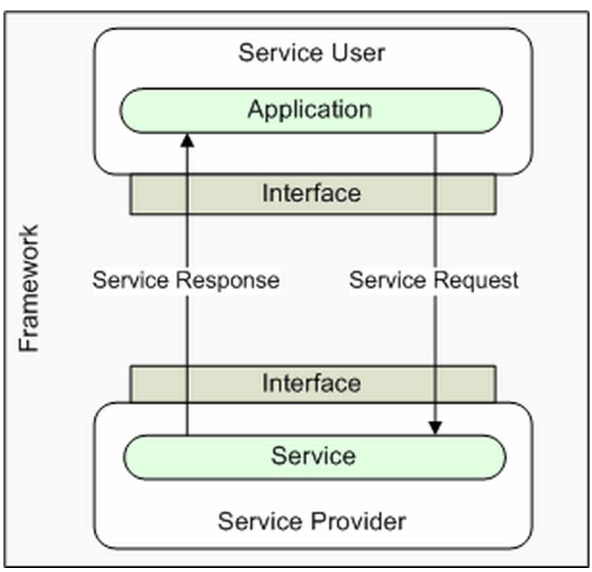
Service-Oriented, Grid Architecture, Publish/Subscribe, and Data Warehouse

This section reviews four software architectural styles: Service-Oriented, Grid Architecture, Publish/Subscribe, and Data Warehouse. To begin, the service-oriented architecture (SOA), as shown in Figure 3, is an architectural alternative which serves to construct extensible and inexpensive software support for business processes and workflows to supply a framework for administration and combination of software services which reflect processes. The services are loosely coupled, which means that they communicate over platform independent interfaces and assume their communication partners to be black boxes. Loose coupling also means that a component is self-sufficient except for its awareness and usage of the other components. Components offer their functions to other components in the form of services, which are similar in concept to publishing methods of an application programming interface (API). Services hide technical details (black box) and are defined functions which can be used on their own or as part of a larger task. The components connect via a mechanism which allows them to be aware of other components and their services while hiding the details of the component communication.

SOAs can be realized based on web services, which are defined by the World Wide Web Consortium (W3C) as "a software system designed to support interoperable machine-to-machine interaction over a network" (Haas & Brown, 2004). Web services are commonly realized with HTTPS (Hypertext Transfer Protocol Secure – a basic protocol for secured passing of web service data from machine to machine over the web), XML (Extensible Markup Language – a widely accepted standard for information exchange), WSDL (Web Services Description Language – a language used to describe the functions of a web service in a machine-readable way, which allows the programmatic localization and utilization of a web service), SOAP (Simple Object Access Protocol – a lightweight communication protocol for message exchange and remote procedure calls which builds on HTTP and XML), and UDDI (Universal Description, Discovery and Integration registry – a registration directory for web service environments). Comprehensive introductions to all mentioned technologies are provided by the W3C (W3C, 2008).

SOAs manage the data of the participating components based on the federated/replicated approach. To manage a healthcare SOA, there also has to be a central component that provides a set of lookup services: a registry for medical services, which stores references to the services that can be used through the SOA (may be based on WSDL and UDDI); a patient identification mechanism, which is effectively a master patient index (MPI) for identifying individual patients across the participating

Figure 3. Service communication in a service-oriented architecture. Service users and providers interact in a decoupled fashion by invoking services through platform independent interfaces. The interaction is embedded into a shared framework.



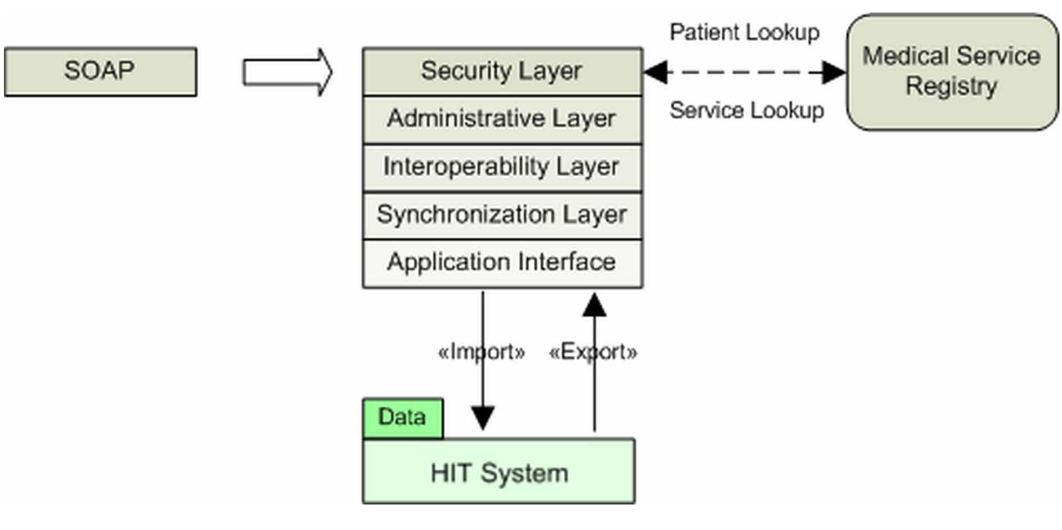
HIT sources; and, a medical record lookup facility, which stores of location of data, the patients index, and meta-data about the nature of the stored patient data (e.g., laboratory results, filled prescriptions, EHR, etc.) along with administrative data (e.g., creation date, last update, etc.). The meta-data will also contain information about the syntactic and semantic formats of the stored record; this allows accessing HIE participants to determine whether they can process the information automatically or not.

The lightweight central architecture means that each participating HIT source needs to implement an interface towards the SOA that is structured as depicted in Figure 4. The security component communicates with the central component of the SOA for authentication and authorization clearance. The administrative layer creates logs and audit trails for satisfying legal requirements. Transformations for semantic and syntactic interoperability are executed by the interoperability layer based on the meta-data provided with every medical record passed through the SOA. The synchronization layer is responsible for buffering or saving data for local use, signalling updates to other SOA participants, and reacting to data updates signalled from the SOA. Finally, the application interface interacts with the underlying HIT source and triggers import and export of data as well as data processing. Clearly, there is a significant set of layers that must be provided for an HIT source to successfully utilize SOA.

Second, to address performance issues in SOA, for complex science and research environments with high resource demands (e.g., CPU cycles, data storage, etc.), grid computing can be utilized. Grid computing describes sharing otherwise unused resources in a cluster (grid) of independent computing nodes.

This happens in a transparent way, in which the additional resources from the created virtual super computer are available, just as electricity is in a power grid, to each of the connected nodes. (Foster, 2002) defines the grid system based on following characteristics: “A Grid is a system that coordinates resources that are not subject to centralized control, using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service.” For the purpose of this paper, grid computing is a realization of SOA with a selection of extremely fine-grained services. While the coarse-grained services of a SOA seem a more suitable solution for the workflow-oriented medical domain, there are emerging initiatives to utilize grid computing for healthcare improvement, medical research, and collaborative care (caBIG, 2004; World Community Grid, 2004), including: medical image processing and analysis, pharmaceutic research/development tasks, complex modelling and

Figure 4. Logical components encapsulated in the HIE interface of HIT sources. Import and export of data between the HIT and the HIE has to pass through a series of conceptual layers of the interface (security, administration, interoperability, synchronization, application). Global lookup of patients and services within the HIE is accomplished by interaction with the medical service registry. Messaging between HIT systems is encapsulated in the SOAP protocol.



visualization jobs, and genomic applications. As personal genomics moves into the forward with genetic information linked to medical patient data, grid computing may be necessary to handle the potential higher volume of data.

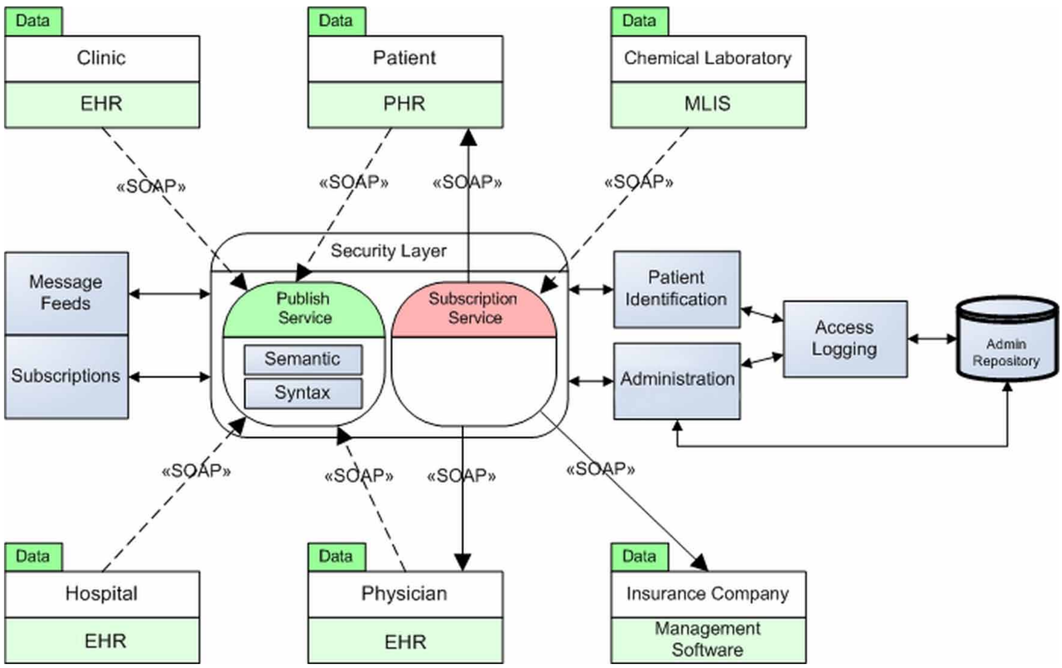
Third, the Publish/subscribe architecture is an asynchronous messaging paradigm describing the relationship of senders (publishers) and receivers (subscribers) with: publisher components which send out categorized messages containing relevant data without a specific target address; subscriber components which allow a subscription to one or several feeds that cover message classes (without knowledge on publishers) and process all received messages according to their needs; and, broker which are optional components for mediating between publishers and subscribers by encapsulating the classification of messages, the subscription process, and the message dispatch. In an HIE system that connects multiple HIT source(s), a publish/subscribe architecture could be used for: the exchange of arbitrary healthcare data between the stakeholders of the domain, health status and advisory notifications such as epidemic alerts, and feedback mechanisms such as drug reaction reporting.

The publisher/subscriber architecture could be applied to healthcare to provide data from multiple HIT source(s), as shown in Figure 5, by implementing the federated data storage approach without central data storage using an enterprise service bus (ESB) for the broker. ESB allows message-based communication and integration between components connected to the bus. This is achieved through a set of services provided by the bus, such as routing services to pass messages from a sender to one or several receivers, and transformation services which can encapsulate syntactic and semantic transformations that are needed in healthcare. Using an ESB for healthcare allows for a higher level of central governance and interoperability control, since all of the communication happens over one component, while keeping the central architecture relatively small. Several ESB products, mostly based on sets of various middleware components, are available from different vendors, in both open and closed source commercial distributions (OpenESB, 2012; WebSphere ESB, 2008; BizTalk Server, 2006; Oracle ESB, 2012; Apache ESB, 2008). Publish/subscribe for healthcare, as depicted in Figure 5, contains the following components: Patient Identification implements an MPI to identify patients across HIT source(s) by creating an index for a given patient with each source responsible for its correct usage; Administration and Access Logging contain the addresses of the registered HIT sources, their meta-data, and access rights information to log which HIE participant received which messages in the publishing process and stores records to meet legal audit requirements; Message Feed Administration and Subscription Administration that contain a list of all message feeds and their current subscribers; Publish Service that is the communication interface for publishers, contains syntactic and semantic transformation services for a central interoperability control, and assures that messages distributed over the ESB are readable by all related subscribers; and, Subscription Service that is the communication interface for subscribers, which notifies all of the subscribers of a message feed on the arrival of new related messages and maintains the messages until they are delivered to the subscribers.

In summary, the publish/subscribe architecture implements the federated approach to data storage (the broker does not persist any data after a successful dispatch), and shares the basic advantages and disadvantages as in the case of SOA. Two potential problems are the unnecessary data replication across the system and the balance which the HIT source(s) must keep between (a) signing up for too many feeds (risk of information overflow) and (b) signing up for too few (loss of relevant information). Too high a number of feeds may slow down the receiving system and have negative impact on patient care; too few feeds may mean a healthcare provider has incomplete data and cannot make a diagnosis or a researcher arrives at an invalid conclusion in a study. The central broker component of the publish/subscribe architecture can remain relatively lightweight and inexpensive, yet still incorporate a good handling of logging, access rights management, and interoperability control.

Lastly, a data warehouse collects data from multiple sources to provide a uniform view on data for querying, analysis, and decision making tasks. Data warehouses (Inmon, 2005) are data collections with the following key characteristics: subject-oriented which describes the way that data

Figure 5. Publish/Subscribe HIE. Participants of the HIE utilize a publish service (to broadcast their data) and a subscription service (to sign up for message feeds and to receive data). The main services are supplemented by a set of auxiliary components to enable the functionality: message feeds and subscriptions are registered, patients and participating systems are traced, and centralized access logging is recorded. Messaging is based on the SOAP protocol.



for the warehouse is chosen, where a possible subject for the healthcare domain would be the patient or the physician; integrated which refers to the common schema in which the data (extracted from heterogeneous sources) is stored; time-variant which refers to the long-term storage of data, allowing analysis related to time; and, finally, non-volatile which means that once stored, data remains in the data warehouse (i.e., there are no delete or overwrite operations). For example, in healthcare, a data warehouse for emergency room patient data from hospitals throughout a country could be utilized to identify and track diseases, epidemics, etc.

Establishing a data warehouse for healthcare data includes two main tasks: extract and integrate data from multiple HIT source(s); and, make the integrated repository available to all eligible HIE participants via a query interface. Data extraction from HIT source(s) occurs periodically through scheduled pull operations (ideally in the after office hours to minimize impact on performance of systems used for providing care) or as push operations initiated by the sources (e.g., when low system load is detected). The extraction and integration process is complex and requires the following subtasks (Inmon, 2005): converting the data into a common format (e.g., HL7 CDA (HL7 CDA, 2007) or CCR (CCR, 2012) in healthcare) with syntactical and semantic checks as well as by transformations as needed to match the utilized storage formats; cleaning the data of irregularities such as data entry errors, e.g., heights in meters instead of feet and inches; integration of the different data sets to suit the data model of the data warehouse, e.g., clearly identify the same John Smith in all sources; and, transformation of the data through summarizing and creating new attributes, e.g., aggregating certain healthcare data that may make an automated clinical decision such as finding multiple high blood pressures for the same patient over time with no appropriate medication prescribed. Note that these four steps are often a semi-automated processes requiring significant human interaction and intervention which is a negative for usage with healthcare. The healthcare data in a warehouse must separate identity with encrypted storage of patient identifiers for safe retrieval of anonymous data and identification of individuals for authorized entities, e.g., for contacting a suitable patient cohorts for research studies.

The data warehouse has advantages that include room for optimization, acceptable and predictable performance, and administration of security and interoperability matters under one governance structure. However, the actuality of a real-time data warehouse might be not realizable in a satisfyingly performant way for a regional or country-wide HIE. For example, in healthcare, there is a need to have high availability of data as it impacts patient care, which is not as critical for an e-commerce application. This will necessitate frequent uploads and synchronizations which may impact HIT source(s) performance; if not, the data would be worthless to the treating medical provider. Further, a country-wide data warehouse may simply not be feasible with the large scale of systems to gather data from and the resulting volume of data. The usage of data warehouses for healthcare may be limited to non-patient care situations but it might be possible to construct very large scale warehouses for offline medical and healthcare data analysis.

Cloud Computing and FHIR

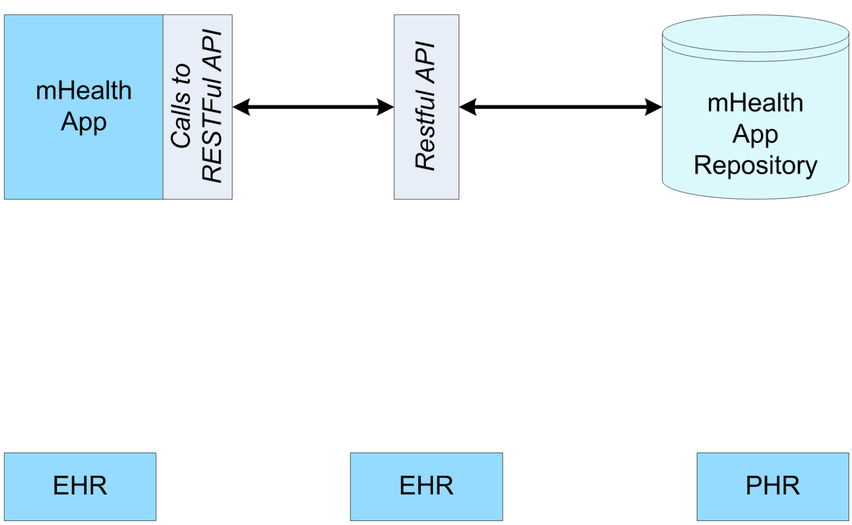
This section explores cloud computing and its relationship to FHIR. The formal definition of cloud computing (Mell & Grance, 2011) sums up the paradigm as follows: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". Cloud computing is an architectural concept which allows the usage of abstract resources that hide computational details and complexity from their users, much like SOA. A key feature of cloud resources is their rapid elasticity (Armbrust et al., 2009) that allows to grow and shrink capacities according to requirements that potentially change in real time, catering towards application fields like e-commerce with very high punctual peaks in infrastructural requirements (e.g., sales on Black Friday or Cyber Monday). Cloud computing typically provides "on-demand self-service" (Armbrust et al., 2009), which means that adjustments to the service (e.g., registration, configuration, extension) are mostly automatic and can be executed programmatically. Service will usually depend on broad network access through standard mechanisms (e.g., web service technologies) which requires a constantly available network connection with sufficient quality of service. Resource pooling and virtualization plays an important role in the background of a typical cloud computing service. However, the user deploying a service to the cloud or using it through a thing client (e.g., a smartphone) has, in most cases, no control of any physical resource instances and is protected from the complexity of that process.

Cloud-computing solutions for enabling health information collection, exchange, and collaboration are appearing in the marketplace, e.g., Microsoft Cloud Services for Health (Microsoft Cloud Services for Health, 2015), the IBM Collaborative Care project (Andrews and Mack, 2011), Logiworks Healthcare Solutions (Logicworks, 2015), VMware vCloud (VMWare vCloud, 2015), etc. Since cloud computing platforms typically foster modular approaches, they facilitate configurable plug-in architectures with support of various extensions for healthcare tasks: decision support, note processing, voice recognition, and patient education. Furthermore, the versatility of the deployment model is capable of streamlining the realization of HIE and the integration of multiple HIT sources by installing regional clouds and enabling communication through hybrid clouds. For example, cloud EHRs such as Practice Fusion (Practice Fusion EHR, 2015) could be used to efficiently implement the application service provider model as described by (Wager, Lee, & Glaser, 2009), which is important in the context of implementing EHRs in small practices that do not have the financial means to hire technical know-how for software and hardware maintenance. A third advantage is that building the virtual patient chart, as introduced earlier in this paper, could largely benefit from a platform with some of the characteristics provided by the cloud services. There are some disadvantages. While HIT sources may implement federation/centralization on top of the cloud paradigm, their foundation is an abstract system under the governance of the cloud provider. This has two results: negative implications for regulatory issues since cloud providers do typically not specify where data centers are located, so that patient data may be stored outside of the patient's country and become subject to foreign laws; and, uncertainty for security particularly if sensitive data become accessible outside of an HIT source through shared hardware and faulty instance encapsulation.

In the second part of this section, we enumerate a number of different mHealth Integration Options to allow an mHealth application to send/receive data with multiple HIT systems via FHIR through the creation of FHIR servers. This scenario is exemplary for the connection of cloud hosted HITs through the exchange of FHIR resources. The scenario includes an arbitrary end-user application (typically a slim implementation on a mobile device) that provides a valuable use-case based on healthcare data fetched from an app-specific repository. The goal is to widen the scope of the mHealth application by making the data of several cloud hosted systems easily available for use. While cloud technologies can be utilized for dramatically simplifying the setup, hosting, and maintenance of systems for healthcare data processing, FHIR with its focus on a simple RESTful API implementation and data representation in JSON and XML is a strong candidate for enabling interoperability towards systems such as an mHealth application, which will typically already make use of those technologies. Conceptually, this can be achieved through a series of different approaches (Baihan et al., 2017).

To begin, Figure 6 contains the architecture components of an mHealth app that includes the app, the app's RESTful API, and the app's repository along with three external HIT Systems (two EHRs and a PHR). The components in Figure 6 will be utilized to define three architectures that illustrate the different ways that the mHealth app can be integrated with the HIT systems based on four issues concerning the mHealth app: overall architecture (e.g., one-tier, two-tier, and three-tier architecture), involved technologies (e.g., RESTful APIs, programmatic APIs, database API), source code availability (i.e., the option to implement changes to of the existing app), and allowable access to HIT (i.e., access methods to the external HIT systems through means such as RESTful APIs, programmatic APIs or cloud services). Note that to make the figures readable, we only show a generalized version with two EHRs; in practice there may be multiple EHRs and other HITs that the mHealth app with interact with. Specifically, we propose: a Basic Architecture that includes a FHIR server that works directly with the mHealth App repository and FHIR servers for the EHRs and the PHR; an Alternative Architecture that includes a FHIR server that works directly with the mHealth

Figure 6. mHealth app and HIT systems. The mHealth app executes calls to a Restful API to interact with its specific repository. The architectural challenge is to enable interaction with external EHRs and a PHR.



App RESTful API and FHIR servers for the EHRs and the PHR; and a Radical Architecture that removes the repository and has FHIR servers for the mHealth App API, the EHRs, and the PHR.

The Basic Architecture option is shown in Figure 7, where the assumption is made that: we have direct access to the mHealth app repository; and, the source codes of mHealth app, mHealth RESTful API, EHRs and PHR HIT systems and their APIs are available. At the bottom of Figure 7, there are FHIR severs (labeled "HIT FHIR") to load/store data from the EHRs and the PHR using their APIs (a third tier) into selected FHIR Resources. Correspondingly, at the top of Figure 7 there is a mHealth.FHIR server to load/store data from the mHealth repository. Basically, each involved HIT systems requires a FHIR server to extract data to/from HIT via FHIR resources that in turn interacts with the mHealth.FHIR server of the mHealth App repository. Interactions from the mHealth app via its RESTful API and from the mHealth RESTful API to the mHealth repository are not impacted. However, to enable the mHealth app to take advantage of HIT systems two new FHIR services mHealth.FHIR.LOAD and mHealth.FHIR.STORE are proposed. The mHealth.FHIR.LOAD service will retrieve all of the data from either the EHRs or the PHR in the FHIR format. This mHealth.FHIR.LOAD service will take the JSON FHIR from the HIT FHIR server and add them into the mHealth repository via an mHealth FHIR service, which converts the FHIR format into mHealth repository format. This will allow all of the mHealth RESTful API calls to use this temporary data. The mHealth.FHIR.STORE service will grab the data from the mHealth repository, via an mHealth FHIR service, which converts the mHealth repository format into the FHIR format and adds them into the two EHRs or the PHR repository. The mHealth.FHIR.LOAD and mHealth.FHIR.STORE services require source code availability of the repository to make the needed calls to stage data back and forth from HIT sources. Note that the mHealth.FHIR.LOAD and mHealth.FHIR.STORE services may also be periodically called to ensure that the repositories at both sides are updated. In this way, the mHealth app, API, and repository are not modified.

In the second option, shown in Figure 8, the situation is similar to the basic option in Figure 7, except that there is no direct access to the mHealth app repository. So the mHealth.FHIR server on the mHealth side is moved to directly interact with the mHealth RESTful API. The HIT FHIR servers for EHRs and PHR are still present as in Figure 7. In this option, the mHealth App continues to use

Figure 7. Basic architecture with direct database access. The HIT systems are extended by FHIR servers to interact with a FHIR server that extends the app repository. The HIT systems become synchronized with the app repository (by periodic synchronization calls). The app and its Restful API remain unchanged.

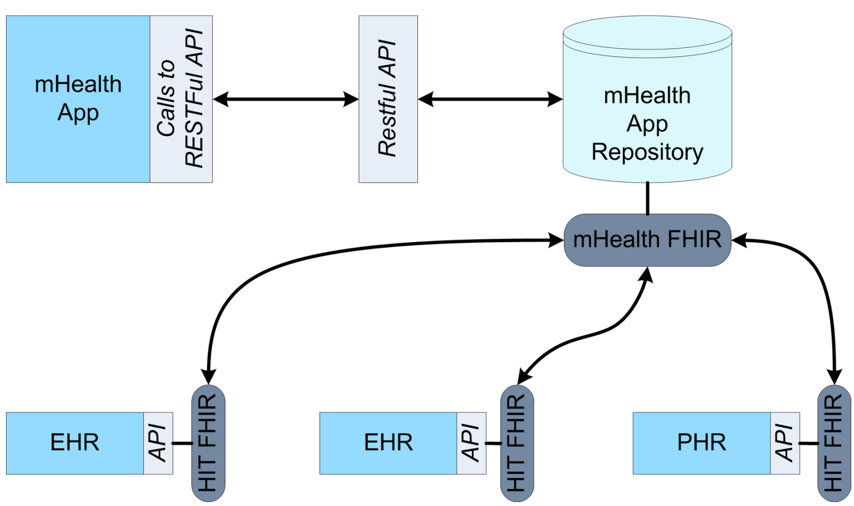
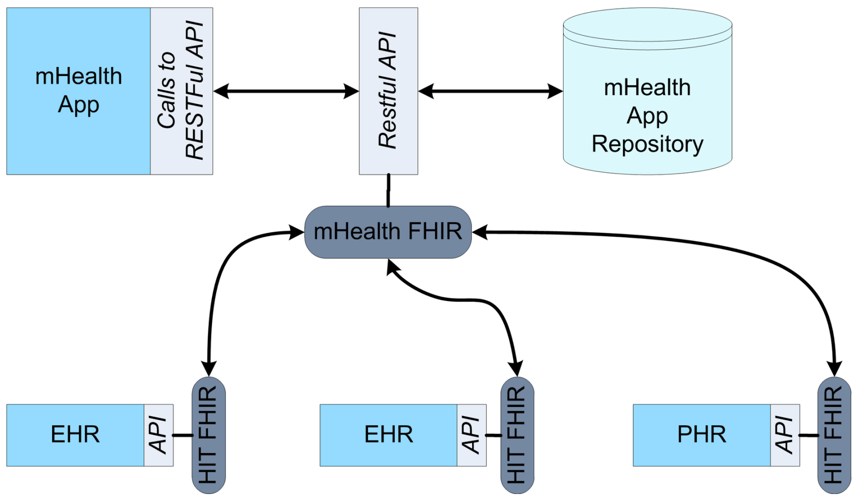


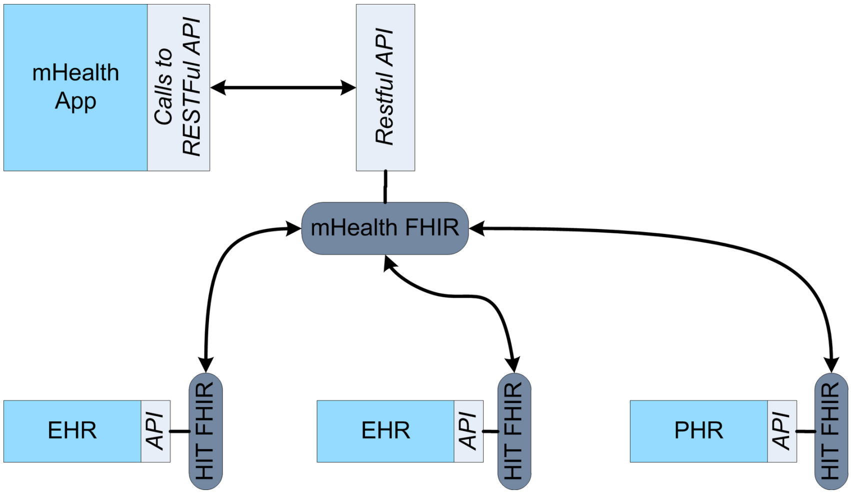
Figure 8. Alternative architecture with mHealth RESTful API access. The HIT systems are extended by FHIR servers to interact with a FHIR server that extends the app's Restful API. The HIT systems become synchronized with the app repository (by on demand synchronization when the Restful API is called). The app and its repository remain unchanged.



the mHealth RESTful API without change, however, the mHealth.FHIR.LOAD and mHealth.FHIR.STORE services now become part of the mHealth RESTful API. That is, each mHealth.FHIR.READ service of the mHealth RESTful API first calls the mHealth.FHIR.LOAD service, which takes an ID of the queried instance and: retrieves the related data from the EHRs or the PHR via their FHIR server; and adds retrieved data into the mHealth repository via another mHealth FHIR API service (i.e. a mHealth.FHIR.CREATE service). This requires slight programmatic changes and source code availability. After that, the mHealth.FHIR.READ service of the mHealth RESTful API will retrieve the related data from the mHealth repository (which is updated with the new data from the HIT system). Similarly, each mHealth.FHIR.CREATE service of the mHealth RESTful API first adds the related data into the mHealth repository. After that, the mHealth.FHIR.CREATE service of the mHealth RESTful API calls the STORE service, which takes the newly added data from the mHealth repository via another mHealth.FHIR API service (i.e., a mHealth.FHIR.READ service) and adds them into the EHR or the PHR databases, via their FHIR service. Note that in this way the mHealth app and its calls to the mHealth.FHIR.RESTful API are not modified; however, a single call to either the mHealth.FHIR.LOAD or mHealth.FHIR.STORE services is added to each RESTful API. This requires source code availability of the RESTful API.

In the Radical Architecture, shown in Figure 9, the connection to the EHRs and the PHR is the same as in the alternative option of Figure 8, but the mHealth app's components are radically altered by removing the repository. As a result, this option has to replace the mHealth app's private data store and instead fully rely on the HIT systems. This option would move and reconfigure all of the mHealth app data so that it could be stored and managed in the HITs. This will require a fundamental rewrite of the code for the mHealth RESTful API while still keeping the interface signatures unchanged to not impact the mHealth app. In this case, every rewritten mHealth RESTful API service will implement a mHealth.FHIR service to directly call the HIT FHIR servers of the EHRs or PHR as required to load/store data as needed. In the Radical Architecture, the services defined are: mHealth.FHIR.CREATE, mHealth.FHIR.READ, mHealth.FHIR.UPDATE, and mHealth.FHIR.DELETE. Source code availability and changing the APIs may be required. This approach is clearly time and effort prohibitive.

Figure 9. Radical Architecture without a database. The HIT systems are extended by FHIR servers to interact with a FHIR server that extends the app's Restful API. Interaction of app and HIT systems is implemented directly without dedicated repository. HIT systems are required to support all of the operations needed by the app. The app itself remains unchanged.



A REGIONAL HIE SCENARIO

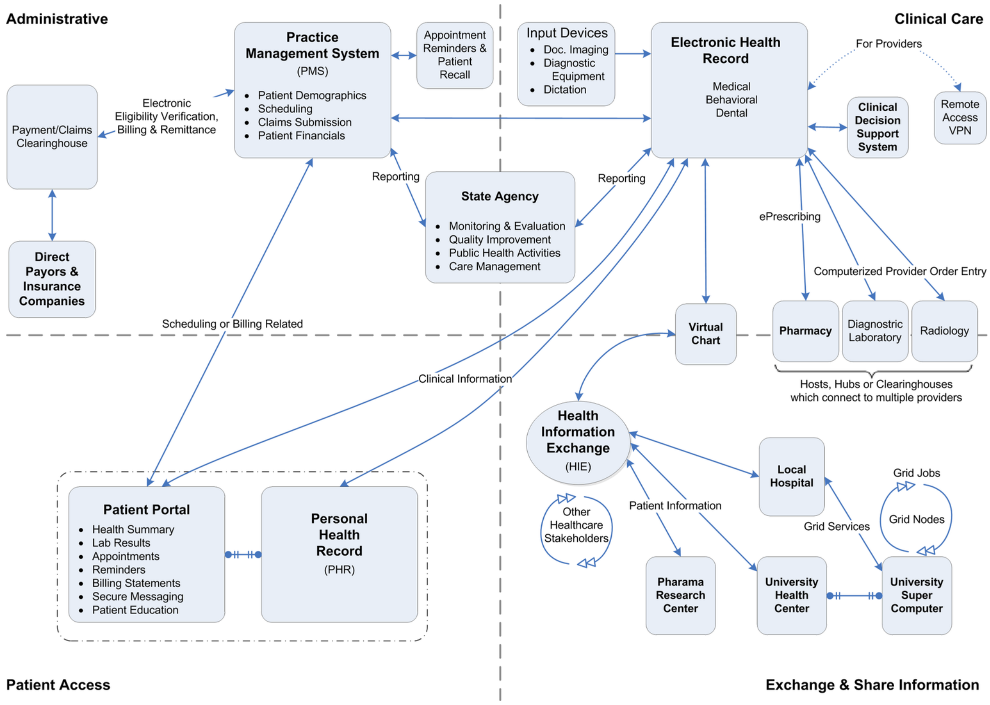
To define the canvas against which the features and components of the proposed hybrid HIE architecture will be explained in the latter portions of this paper, this section introduces a set of HIT sources that participate in a regional HIE scenario, lists the employed HITs, and identifies selected examples for collaboration between sources. For the purpose of this discussion, a source is classified either as a data supplier or as a data consumer. Note that to maintain clarity, a simplified model is used, with a limited amount of sources and under the assumption that each source fits exclusively into its category. In an actual HIE, suppliers naturally will consume data at the same time, and vice versa. In order to place the data suppliers and consumers into a concrete context, Figure 10 illustrates a real-life regional HIT system infrastructure. The example is centered on the HITs of a physician's practice (e.g., a community practice) and explores the support for administrative tasks, clinical care, patient access, and the exchange of information in the process of providing patient care.

The core of the HIE infrastructure showcases the use of a selection of HIT sources as discussed earlier in this paper: Administration of the practice, to manage patient demographics, appointments, billing tasks, etc. are supported by the patient management system (PMS), which interfaces with insurance companies and other payers. Clinical care is provided with the support of an EHR system in combination with a decision support system and various input devices; PMS and EHR automatically report to healthcare-related agencies. The EHR supports an e-prescribing feature as well as electronic communication with various medical laboratories. Patients are able to access information from the practice's systems via a web-based patient portal and the PHR. The EHR system accesses information from other external providers via HIE to access a hospitals EHR or a laboratory system, and allows reviewing a patient's medical history via a virtual chart. Note that the parenthetical notation in the remainder of this section is referring to the location of each HIT source in Figure 10.

To begin, the data suppliers that are shown in Figure 10 are reviewed:

- **Community practice (upper half):** A medical practice operated by several physicians (e.g., a general practitioner, a pediatrician, an internist, and a radiologist) and their staff. The HITs used in the practice are a decision support system (DSS), a practice management system, a web-

Figure 10. Overview of a HIT system infrastructure. The infrastructure is centered on the HITs of a physician's practice. Administration tasks are supported by the patient management system (PMS), which interfaces with insurance companies and other payers. Clinical care is provided with the support of an EHR system and a decision support system. PMS and EHR automatically report to healthcare-related agencies. The EHR e-prescribing as well as electronic communication with various medical laboratories. Patients have access to data from the practice's systems via a web-based patient portal and the PHR. The EHR system shares information from other external healthcare providers and research facilities via HIE and virtual chart.



based patient portal and an EHR. The practice collaborates with the virtual chart, pharmacies (e-prescriptions), and state agencies (reporting of selected diseases as mandated by law);

- **Local hospital (lower right):** A hospital providing healthcare to inpatients and outpatients from the local population. The hospital staff (i.e., nurses, physicians, ER personal, etc.) files patient data with an EHR and uses a DSS, where required by patient care. The hospital's radiology department participates in a large-scale breast cancer screening program and stores the resulting images in a custom database (the beginning of a data warehouse) for automatic analysis. The hospital collaborates with a university supercomputer center (automated x-ray analysis), the virtual chart, and state agencies;
- **University health center (lower right):** A research focused healthcare facility with a limited patient cohort (two hundred inpatients), maintaining an infrastructure for clinical studies, and running a human tissue and specimen bank. A medical chemistry laboratory equipped with a medical laboratory information system (MLIS) is also part of the center. The health center staff (i.e., physicians, clinical researchers, nurses, etc.) utilizes an EHR system and custom databases for the management of data related to inpatients and study participants. The EHR system supports automatic drug tolerance reporting to the pharmaceutical research center of a sponsor. The center collaborates with a university supercomputer center (for genomic research), state agencies, and a pharmaceutical research center;
- **Personal health record (lower left):** A web-based application used by patients to maintain their medical and health histories, including conditions, received treatments, current and past medications, allergies, food supplements, health markers, fitness data, etc. Significant parts of

the stored data originate from patient entries (manual or recorded through smartphone apps or other consumer-grade tracking devices such as wearables). The application also supports a calendar function for the scheduling and planning of practice visits. The record is connected to the virtual chart.

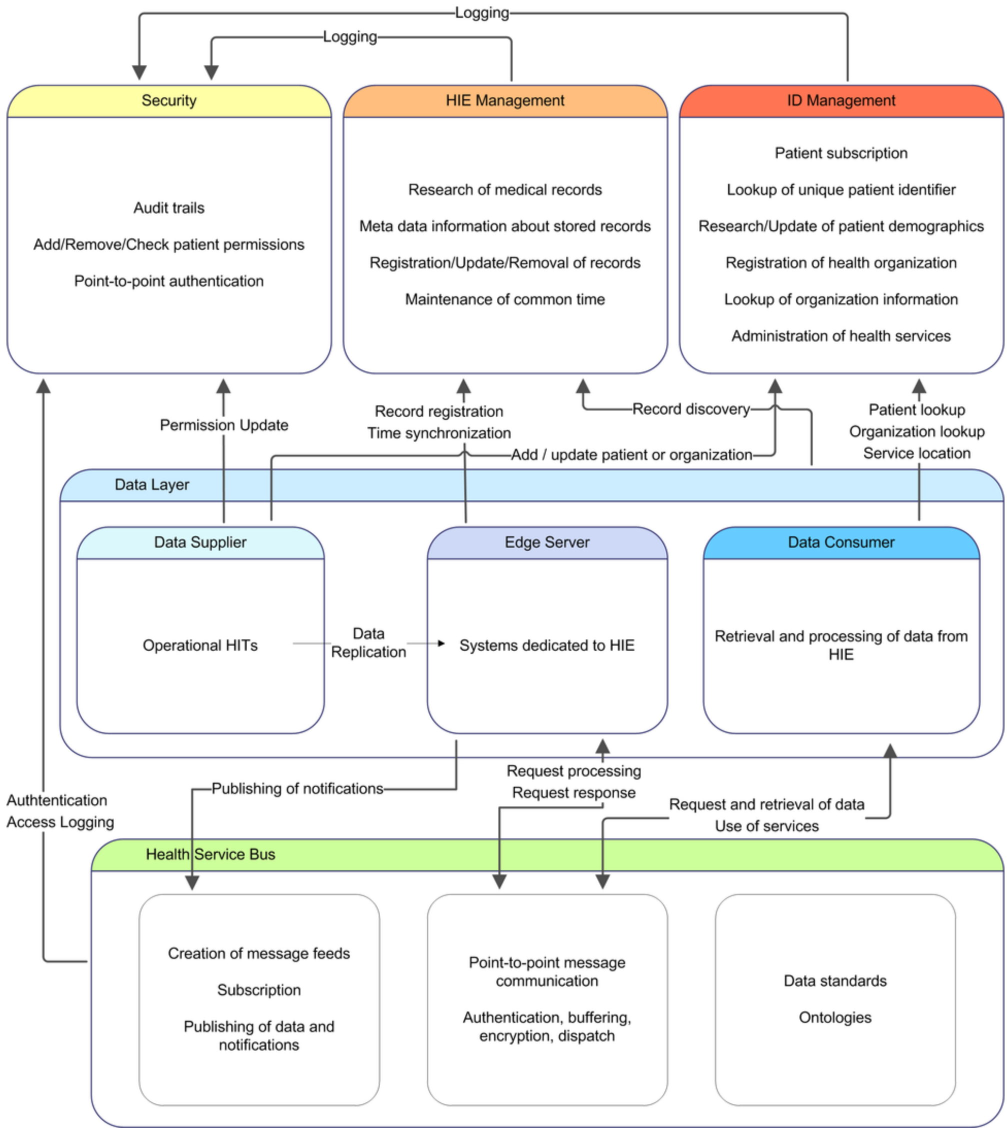
To complete the discussion, the data consumers that are shown in Figure 10 are reviewed:

- **Local pharmacies (middle right):** A group of pharmacies offering a variety of over-the-counter and prescription drugs. The main HIT used is a pharmacy management system (management of prescription histories, e-prescribing interface, business processes). Each pharmacy is connected to the community practice (e-prescriptions), the local hospital, and the university health center;
- **State agency (upper middle):** Institution such as the Department of Mental Health and Addiction Services, the Department of Children and Families, or the Centers for Disease Control and Prevention. The agencies provide various healthcare-related services to the population such as issuing reports on health-trends, monitoring of the population-wide health status, executions of screening programs (e.g., alcohol abuse), and tracking of health-related events (e.g., in the case of a pandemic outbreak). The agency collects data generated by the community practice, the local hospital, and the university health center;
- **Insurance company (upper left):** A company providing a variety of health plans for public agencies and private and corporate customers; handles eligibility verification and billing for the local hospital and the community practice;
- **Pharmaceutical research center (lower right):** A research facility operated by a pharmaceutical company. The facility collaborates with the university health center (collection of anonymized data for research purposes) as well as with the local hospital and the community practice (recruitment of patients and retrieval of results from pharmaceutical studies);
- **University supercomputer center (lower right):** A virtual supercomputer operated by the university's center for transitional science. The system is based on a grid architecture, connecting grid nodes at the university and nationwide;
- **Virtual chart (middle right):** An HIE application providing consistent, complete, and historically accurate patient medical records through automatic integration of data from various source repositories. The chart is linked to all HIE participants that process medical histories (such as the local hospital or the community practice).

A Hybrid HIE Architecture (HHIEA)

The architectural alternatives studied in this paper have many disadvantages that make their usage unsuitable for an HIE solution using only one of the alternatives. This section proposes a hybrid HIE architecture (HHIEA) at a design level, by proposing a combination of the studied architectural styles which both balances and mitigates the advantages and disadvantages. The resulting HHIEA establishes a more effective and flexible architecture that facilitates the exchange of information between multiple HIT sources and provides a comprehensive and integrated view on healthcare data. HHIEA combines: data warehouse (Prokosch & Ganslandt, 2009; Zeh, 2003), SOA (Rosen, 2008; Ryan & Eklund, 2008), grid computing (Bilykh et al., 2003; Foster, 2002), and the publish/subscribe paradigm (Eugster et al., 2003; Singh, Vargas, Bacon, & Moody, 2008). Furthermore, it relies on FHIR for the exchange of healthcare data (HL7 FHIR, 2016). The proposed HHIEA, as shown in Figure 11, is presented in five logical groups in separately labeled subsections, namely: the Data Layer where all of the data suppliers and data consumers are logically part of this group; ID Management that is utilized to both identify and differentiate patients and organizations participating in the HIE; HIE Management that is composed of tasks related to the maintenance of medical record references; Security that contains components that are related to the management of audit trails, patient consents,

Figure 11. High level overview of the HHIE Architecture illustrating its components and their interactions. The top three components (Security, HIE Management, ID Management) coincide to the major functionality. The middle three components (Data Supplier, Edger Server, Data Consumer) are the main systems of the architecture. The Health Service Bus provides the means to interact and securely exchange information according to standards and ontologies.



and participant authentication; and, a Health Service Bus that is responsible for ensuring the precise passing messages between the HIT sources. The discussion on HHIEA is completed with a section that utilizes the realistic healthcare scenario in order to fully demonstrate the ability of HHIEA to attain and support HIE.

Data Layer

The data layer of the proposed HHIEA, as shown in the middle of Figure 11, is the location where information that has been extracted from each HIT source is utilized with a replicated storage style with edge servers in order to acquire information from the HIT sources. The processes for the initial loading of each HIT source (data supplier) to the replicas and subsequent incremental updates at periodic intervals are left to the administrators of the deployed HHIEA. If data of a HIT source is requested by a data consumer then it is served from the dedicated edge server to which the HIT source replicates its contents. The physical location of the replica can be at the HIT source or at a neutral location, where data from all source(s) is assembled; the location is transparent to all users. As shown in Figure 12, the data suppliers in the data layer are intended to provide the appropriate services that are needed to create the initial replica and incrementally update from the HIT source(s). The data consumers in the data layer provide the appropriate services for the stakeholders to access the aggregated information that spans these multiple HIT source(s) that are now replicas in the data layer. As a result, the data layer of HHIEA exploits the replication style to create and organize the replicas of the HIT source(s), leverages the publish/subscribe architecture to stage data in and out of the replicas, and provides data consumers with the means to make use of health information (e.g., for aggregation in a data warehouse).

This approach to the data layer insures that governing HIE participants have full control over which data is shared via HIE while simultaneously decoupling performance and security issues from the operative systems of the replicas. The combination of replication and publish/subscribe into the data layer promotes the definition of security policies, privacy regulations, and permissions at this common layer. In setting up each replica from a specific HIT source, the data owner (e.g., a hospital with an EHR) can determine what information to share, as depicted in Figure 13, where information from the EHR, patient portal, practice management system, etc., can be shared. The edge servers can enable interoperability by allowing them to enforce common standards for storage (i.e., syntactic interoperability) and to manage the stored data with common ontologies (i.e., semantic interoperability).

Towards the goal of easing interoperability, Figure 12 and Figure 13 show an FHIR based approach to data exchange. While the HHIEA is conceptually capable of handling a variety of messaging standards, the use of FHIR comes with significant benefits and flexibility as discussed in previous sections. In Figure 12, the edge servers provide access to the replicated data in the form of FHIR resources, ideally through a RESTful API, which is implemented by the Edge FHIR

Figure 12. The data layer with data suppliers, edge systems, and data consumers. Data suppliers are HIT systems such as EHRs used for provision of care. Data collected at the source systems is moved through a replication process (custom to each source system) to edge servers and therefore physically separated from the operational care delivery. Edge servers are HIE specific uniform data storage systems outside of the organizational borders of the suppliers. They provide standardized access to their data based on FHIR to data consumers in the HHIEA. Data consumers in turn retrieve data (as FHIR resources) from the suppliers for processing, aggregating, analysing, etc.

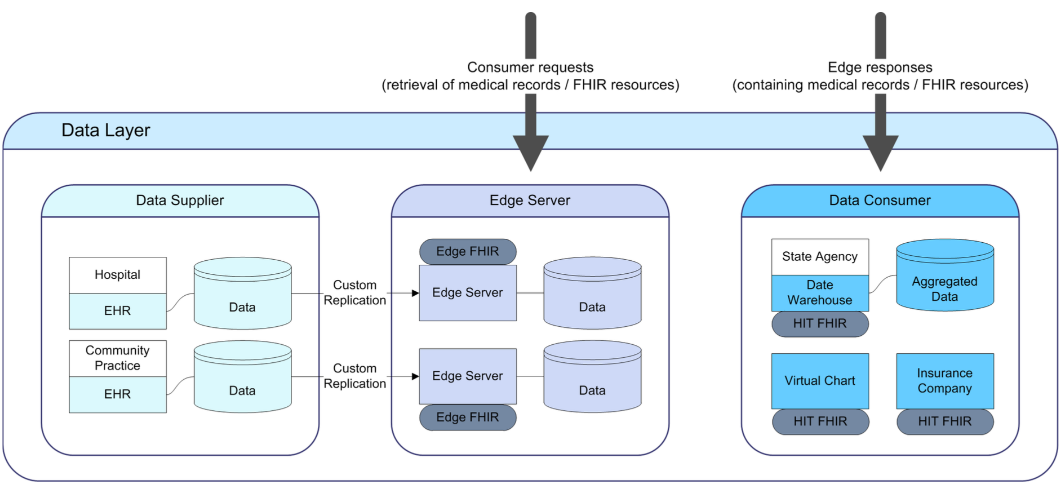
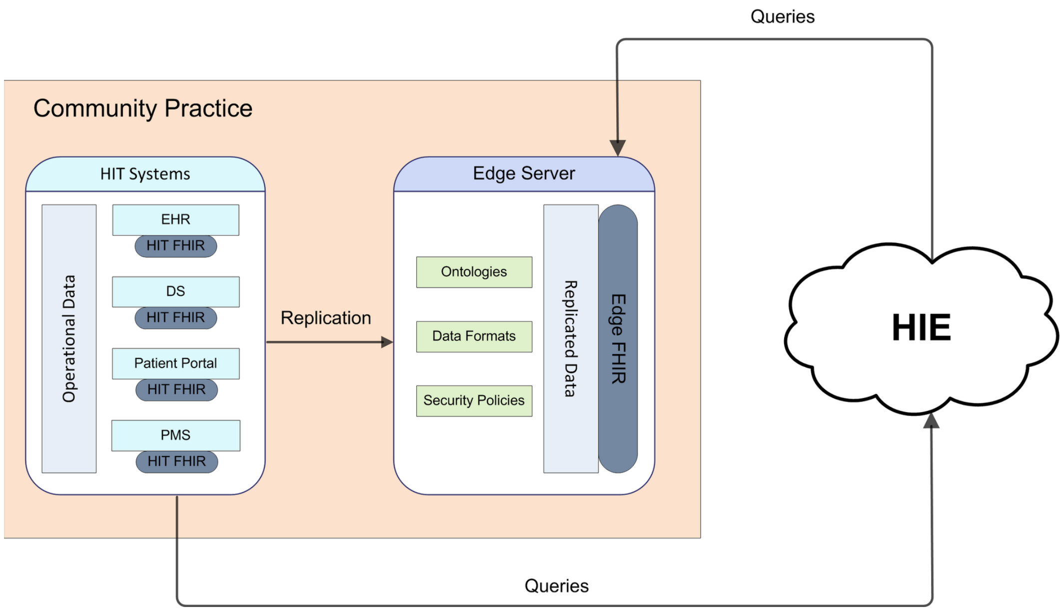


Figure 13. A community practice with edge server system and HIE interactions. Operational data from the practice's HIT systems is replicated to an edge server that accepts FHIR queries from the HIE. The content of the edge server remains under the control of the community practice and is subject to the implemented replication (which has to match HIE wide ontologies and data formats) and local security policies. The local data of the HIT systems can be extended by executing FHIR queries to the HIE.

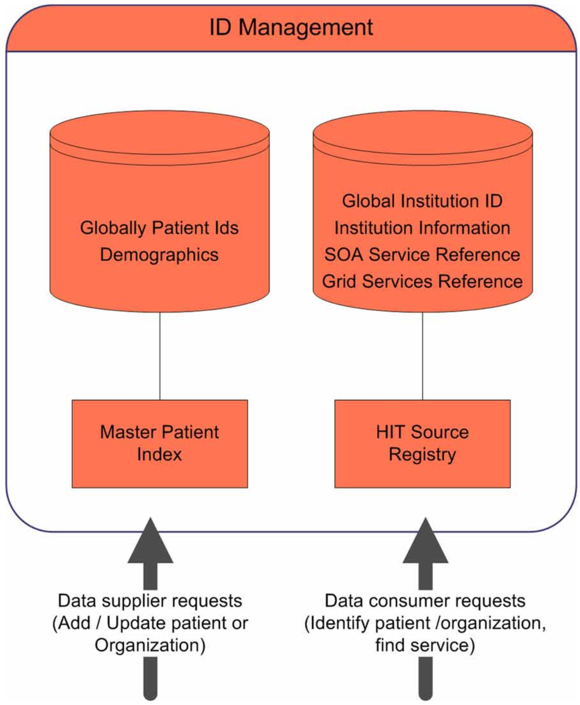


components. The edge servers can be utilized to unburden the operational systems from any additional data transformations that may be required to support the FHIR standard. On the other side of the interaction, the operational systems acting as data consumers have to be extended by HIT FHIR components that are capable of reading and processing FHIR resources that are received as answers from queries to the HIE. The same pattern is used in the detailed view of Figure 13, where HITs make queries to the HIE and receive FHIR resources as response, which are processes by the HIT FHIR components. Simultaneously, the data gathered at the HITs is replicated to the edge server, which makes it available as FHIR resources to the HIE through the Edge FHIR component. Replication to the edge server is solved separately by each HIT source which allows for a high level of customization to meet special characteristics of the legacy systems. Therefore, the process of edge server deployment brings the opportunity to enforce a canonical message format and a common semantic foundation for exchanging data.

ID Management

The components in the ID management group, as illustrated in the upper right of Figure 11, provide the means for HHIEA to maintain cross-organizational identification of HIE participants. For the identification of patients, a master patient index (MPI) component must be implemented, as shown in Figure 14, to allow this component to register a unique ID for every patient participating in the HIE, and store it with a set of demographic information. Furthermore, the MPI must support two types of queries: queries parameterized with a valid global patient ID, and queries parameterized with a set of demographic information (e.g., first name, last name, date of birth, and birth place). The first type returns the demographics for a given patient for identity verification purposes and for updating demographic information (e.g., changed place of residence). With the second type, the unique identifier of a patient can be retrieved for further use (or created if the patient is not registered in the HIE yet).

Figure 14. The Identity Management Component: This component provides the means to: assign a global ID in the form of a master patient index (MPI) to an individual with given demographic information; update the stored demographics; and, optionally execute a retrieval query of demographics for a given MPI and a retrieval query of a MPI for a set of given demographics. The component also provides a registry of institution IDs for HIT sources to allow: registration and update of metadata about participating nodes; retrieval of an ID for a set of metadata as search criterion; retrieval of an institutions metadata for an ID as search criterion; and, retrieval of SOA/Grid services for a participating node.

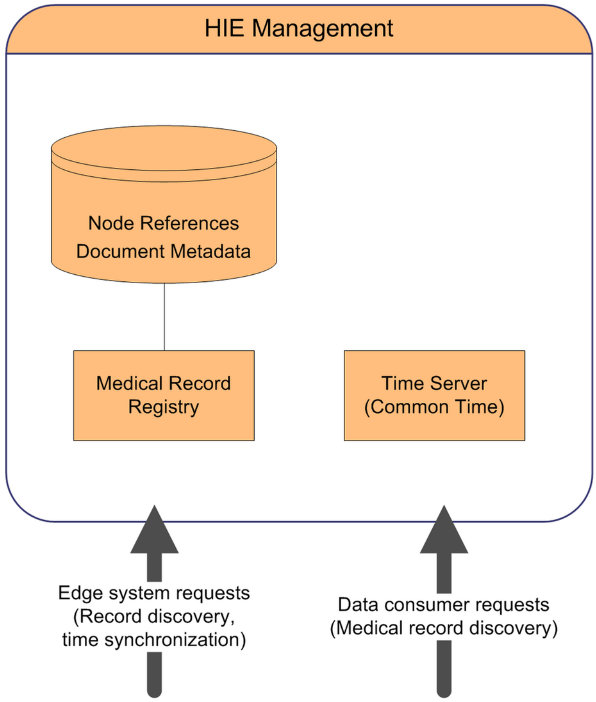


The MPI component will typically be queried by data consumers of the data layer to retrieve the MPI for a patient, which is needed to locate patient records. Data suppliers will access the component to sign up new patients (request a new MPI) and to update existing demographics. A similar component, the HIT source registry, is utilized for the identification of healthcare organizations (i.e., hospitals, clinics, labs, etc.) participating in the HIE. The registry maintains a globally unique ID for each HIT source, which is distributed to an organization during an adequately controlled registration process to avoid service violations and to ensure a secure domain. In addition to the HIT source ID, the HIT source registry stores general information about an organization (e.g., organization type, physical address, etc.) in query language form. Furthermore, each organization can offer SOA and grid services via this component; complete and machine-readable descriptions to those services are also kept in the HIT source registry. The ID Management group will, in most cases, be queried by data consumers for the lookup of services. Data suppliers will access the registry to update information about their organization.

HIE Management

The HIE management group of the proposed HHIEA, as shown in the upper middle of Figure 11, consists of two components: a medical record registry that is capable of mapping the MPI to all of the multiple replicas that may contain data for a patient; and, a global clock for time-stamps to eliminate inconsistencies in data. As shown in Figure 15, the medical record registry identifies medical records across the HIE by processing queries from data consumers; these

Figure 15. The HIE Management Component: This component provides a medical record registry with metadata for the patient records available in the HIE. The registry can be queried using an MPI to retrieve the location (the node storing the record), type, and format of all records for a given patient and tracks addition of new records to the HHIEA. The component also provides a common time server for the time synchronization of participating edge servers.



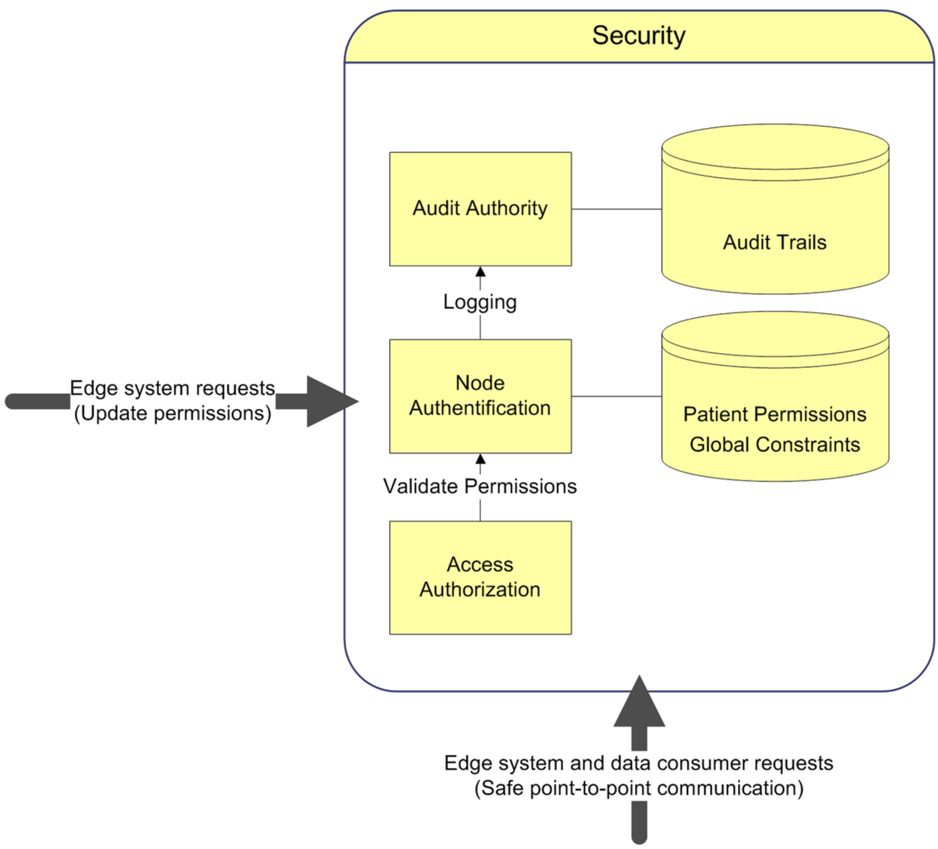
queries must contain a valid MPI value. After receiving a query from an eligible data consumer, the registry will return a list of HIT source identifiers which reference all of the replicas that are storing records related to the MPI in question. For a baseline, a simple retrieval of all available related HIT source identifiers can be implemented that would bring in entire medical records for a patient from the HIT source replicas. A more comprehensive querying approach may be smart enough to limit the retrieval to only recently added records for the patient. With the list of identifiers, a data consumer is able to retrieve the contact information of the data suppliers storing the records and request them via an appropriate service such as RESTful API serving FHIR resources which are finally served by the replica of the data supplier. In addition to the identifier, the registry stores metadata on the data formats available for the record as well as the ontologies necessary to interpret the record's content. Based on this meta-data, a requesting data consumer can determine if the processing of the record can be managed automatically or if it will require human intervention. The medical record registry will be, in most cases, accessed by data consumers during the search for records related to a patient; edge systems will also access the registry and add information each time a new record becomes available or at some periodic interval. The global clock component addresses problems arising from asynchronous/ faulty timestamps in medical records. For example, if two records related to one incident are available on two different replicas and the timestamp of one of them is offset by a year due to a faulty set system time, an automatic interpretation of the records will result in the assumption that the incident occurred twice. To avoid such situations, all HIT sources and their replicas must synchronize their timestamps with the help of the global clock.

Security

In order to enable the exchange of sensitive healthcare data, a security component of the proposed HHIEA is required, as shown in the upper left of Figure 11. Security and access control in the heterogeneous and federated healthcare domain is a highly complex topic, particularly when taking into account additional requirements from mHealth applications (Rivera Sánchez et al., 2016; Sanzi, Demurjian, Agresta, & Murphy, 2016). Thus, implementation details cannot be provided in the scope of this paper. However, to support the key requirements, a security group for the HHIEA containing the set of components illustrated in Figure 16 is proposed: an authentication component, an access authorization component, and an audit authority component. For secure point-to-point message communication between the participating data suppliers and data consumers, means for identity authentication of communication partners must be provided. This task is addressed by the authentication component, which is utilized as a central trusted third party.

In a healthcare setting, a medical provider (e.g., physician) would typically be authenticated to their EHR and have access to an EHR. In HHIEA, the replicas bring together multiple HIT sources, and the likelihood is that the physician has not been authenticated to all sources. As a result, the authentication component must be capable of reviewing a physician's authorized privileges on the HIT sources they are explicitly authorized on (EHR) and utilize this information to determine the other

Figure 16. The Security Management Component: This component implements access authorization for providing point-to-point node authentication for all communication in the HHIEA. This makes use of a node authentication registry, which is capable of adding, removing and checking patient consents for all stored records. All of the processed requests are logged by the audit authority, which stores complete audit trains for legal purposes.



sources (replicas) that would also be accessible. For the authentication component, a solution based on federated business SOA approaches built around certificates and public/private key encryption could be adopted. In addition to authentication, adequate access control is a mandatory requirement for the communication between HIT sources. This would include techniques such as: role-based access control (RBAC) (Ferraiolo, Sandhu, Gavrila, Kuhn, & Chandramouli, 2001) that focuses on the responsibilities of the users for an application per role; discretionary access control (DAC) (Na & Cheon, 2000) to allow both authority and privileges to be passed from one user to another; and mandatory access control (MAC) (Bell & LaPadula, 1973) that assigns sensitivity levels to subjects (clearances) and objects (classifications) to control access to information. RBAC can set up different privileges for different categories of users such as roles for physician, nurse, therapist, etc., that differ in privileges (what can be read and/or written). DAC can allow privileges to be passed among users such as the case where a physician delegates access to the records of their patients to the on-call physician (for nights/weekends). Lastly, MAC can classify patient data, differentiating between more generally available data (demographics, medications, tests, etc.) from very sensitive data (e.g., psychiatry records).

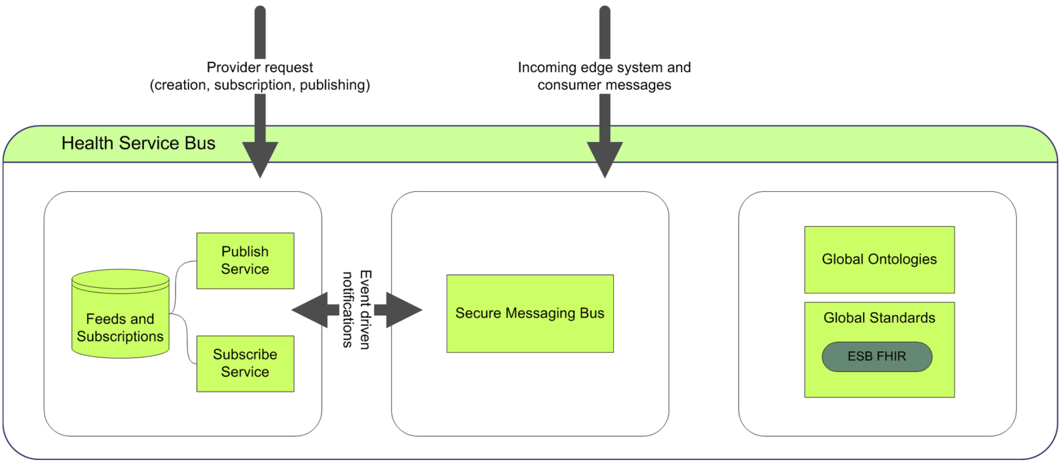
The access to medical records must frequently be checked against permissions given by the patient, requiring the access authorization component to store patient consents. For each request involving patient data, the authorization component has to check if the communication needs to be explicitly permitted, and if this is the case, it has to verify the compliance with the stored consents and global constraints. Furthermore, edge systems will access the component to update stored permissions. In this infrastructure, the security group is the point of HHIEA which is aware of all of the communication across the multiple HIT sources that form the HIE. Thus, the authentication component and the access authorization component are coupled with an audit authority, which records complete audit trails of connections between data providers and data suppliers as well as details about executed data transfers. The audit authority component provides an interface for the retrieval of those audit trails, so that they may be used to meet regulative requirements or for system maintenance and analysis. For example, while two physicians may have the same role, they may be authorized to be in charge of different patients at a hospital. As a result, most hospitals audit access to patient records to insure that their employees have not been accessing patients to whom they have not been authorized (which historically has happened when a famous person is in the hospital). Note that establishing adequate security for communication in the HHIEA has significant overhead and a negative impact on the overall system performance; an estimate on complexity and the costs of this task could be attained by examining a secure multi-enterprise SOA. In addition, the Extensible Access Control Language (XACML, 2003) can be utilized to specify security policies.

Health Service Bus

The health service bus component of the proposed HHIEA, as shown in the bottom of Figure 11, contains components relevant for communication of data suppliers and data consumers. The key component is the secure messaging bus, which enables message-based, asynchronous, point-to-point communication between the participating HIT sources. With the combination of the functionalities of the other HHIEA groups presented in the previous sections, the HIT sources are able to form messages containing all of the necessary information required for the messaging bus. The service bus receives and buffers those messages, clears security and logging issues with the components of the security component, and dispatches the messages to their targets, as shown in Figure 17.

The prototypical secure messaging infrastructure can be based on different approaches such as web services as interfaces for the HIT source replicas, SOAP (Simple Object Access Protocol) as the basic message format, and an Enterprise Service Bus (ESB) solution for business SOA as the connecting element (Curbera et al., 2002). The term ESB roughly describes the concept of a middleware component which allows message-based communication between entities connected to the bus.

Figure 17. The Health Service Bus: This component provides a secure messaging bus for message based point-to-point communication in the HHIEA, which includes authentication, message encryption, buffering, routing and dispatch. The bus enforces message interoperability based on set of stored global ontologies and standards by checking the payload of the passed messages (e.g., FHIR schema conformance). The component also implements a publish service and a subscribe service for the creation of message feeds, subscription to feeds, and publishing of new notification to a feed.



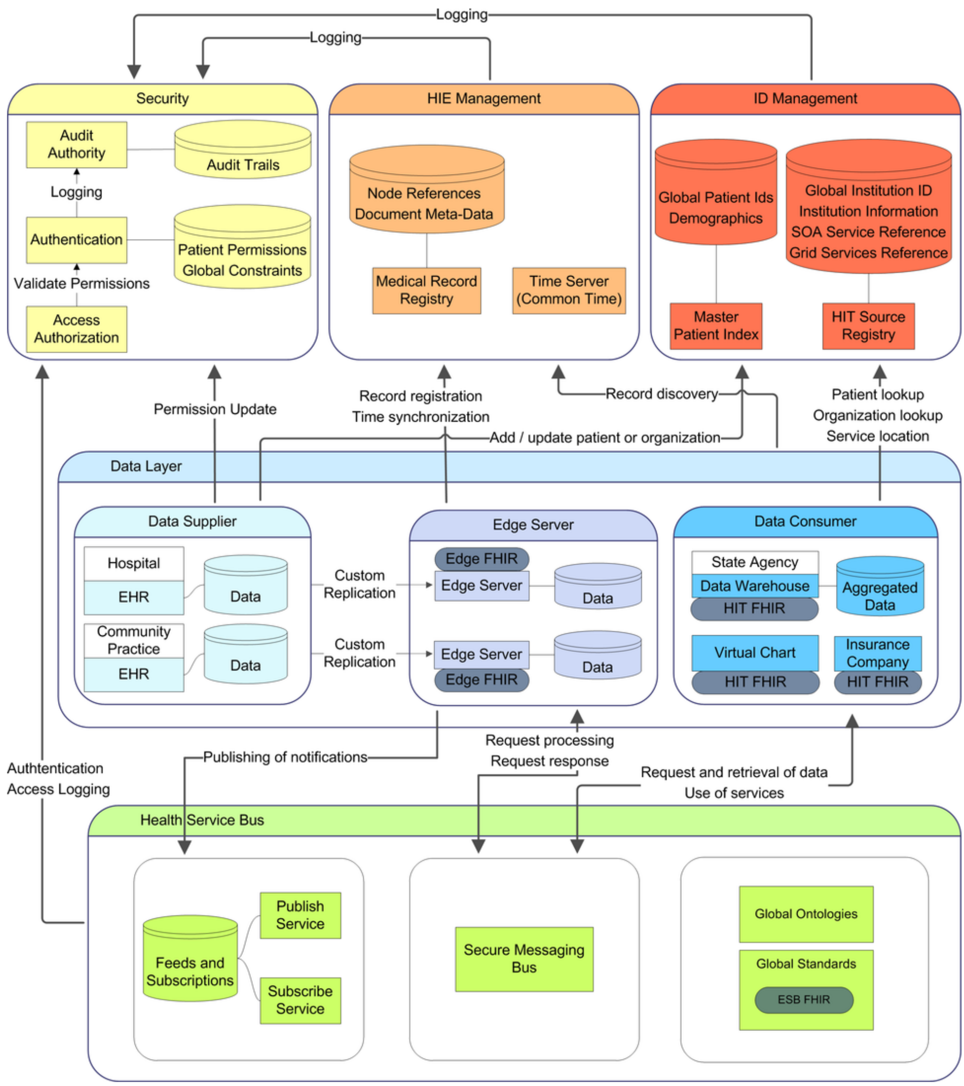
Custom ESB extensions for the support of grid services will be required to minimize security overhead for messages containing anonymized and non-sensitive data. More comprehensive ESBs will require syntactic and semantic interoperability among the replicas through the support of standards and ontologies from the message bus side. In an HHIEA implementation relying on the FHIR standard, the ESB would be enabled to verify that the resources passed as a payload of messages conform to such restrictions as XML schemas or JSON schemas. Furthermore, this is the location to reconcile differences in healthcare nomenclature such as the need to reconcile the various medical terms for heart attack: Myocardial Infarction, cardiac arrest, coronary infarction, etc. This means that the health service bus maintains a set of global standards and provides services for format transformations between those standards. Equally, a global ontology, alongside an ontology mapping engine, can translate between the different local ontologies utilized by the HIT sources. Finally, the health service bus can be equipped with a publisher service and a subscriber service. Both services are connected to a repository storing data related to message feeds and subscriptions, and can be used to implement the functionalities of a publish/subscribe architecture: creation of message feeds, subscription, and publishing of messages via feed (message dispatch can be realized through the secure messaging bus).

HHIEA and the Regional Scenario

This section concludes the presentation of the HHIEA by examining the identified collaborative linking of HIT sources and stakeholders, where data from each HIT source is extracted to form a replica that operates as a data supplier in the HIE, and each stakeholder is interested in accessing patient data across multiple edge servers and operates as a data consumer. The goal is to demonstrate the way that the full HHIEA architecture as shown in Figure 18 can efficiently manage an HIE scenario by providing the combined functionalities of multiple architectural styles. Figure 19 visualizes the process by complementing the previously HIT infrastructure with details from the proposed HHIEA.

HHIEA components in Figure 19 are the publish and subscribe services, which are used to realize automated reporting, and the secure messaging bus, which implements the extendable message-based communication between the various HIT sources and allows for the execution of SOA/grid services. Furthermore, each HIT source is equipped with an SOAP interface for the generation of outgoing messages and the processing of incoming messages. HITs that frequently exchange medical record

Figure 18. A view of the detailed hybrid HIE Architecture illustrating its components, sub-components, and their interactions

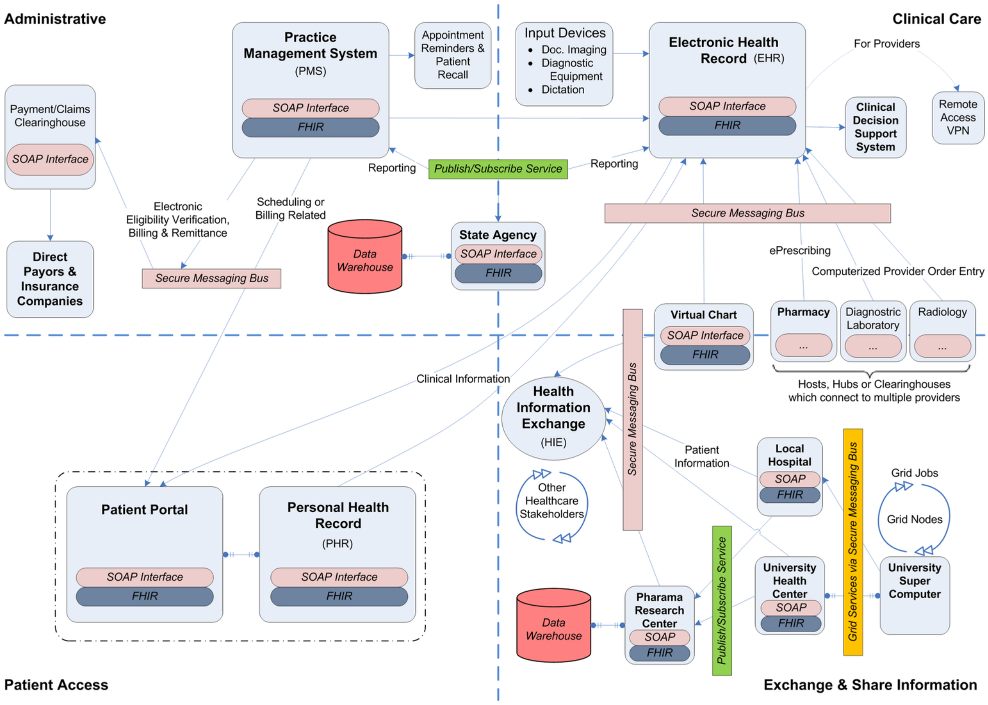


data additionally make use of FHIR server implementations to standardize transferred messages. Data warehouses add support for data aggregation and analysis at eligible HIT sources, and may also be accessed via message bus and through services offered by the HIT source. In this infrastructure, concepts from multiple system architectures are used to resolve identified collaborative links. The following reviews the usage of the various architectures in support of the realistic scenario.

Service-Oriented Architecture

By utilizing SOA, the HHIEA is designed to support service discovery, patient lookup, medical record localization, and secure point-to-point communication. In addition, a full realization of the health service bus includes comprehensive support for interoperability. In combination, these functionalities provide the means for retrieving all of the available medical records related to a certain patient and, thus, for a virtual chart. This resolves the links Community Practice – Virtual Chart, Local Hospital – Virtual Chart, and PHR – Virtual Chart. Furthermore, the SOA functionalities cover all of the

Figure 19. HHIEA applied to the HIT Infrastructure: HHIE components are utilized to enable data exchange and collaboration between systems of the infrastructure. Publish and subscribe services realize automated reporting. A secure messaging bus implements extendable message-based communication between HIT systems and allows for the execution of SOA/grid services. Each HIT source is equipped with an SOAP interface for the generation of outgoing messages and the processing of incoming messages. HITs that frequently exchange medical record data are also provided with FHIR server implementations to standardize transferred messages. Data warehouses at strategic HIT sources (State Agency and Research Center) aggregate and analyse collected data provide access to it via message bus and through services offered by their host HIT systems.



links which rely on the transfer of arbitrary data, such as Community Practice – Local Pharmacy, Local Pharmacy University Health Center, Local Pharmacy – Local Hospital, Insurance Company – Community Practice, and Insurance Company – Local Hospital.

Publisher/Subscriber

The proposed health service bus is equipped with a publish/subscribe service, the infrastructure for the administration of message feeds, and the means to dispatch messages containing arbitrary data. This resolves all of the links related to event-driven tasks (e.g., reporting of medical cases) such as Community Practice – State Agency, Local Hospital – State Agency, University Health Center – State Agency, Pharmaceutical Research Center – Local Hospital, Pharmaceutical Research Center – Community Practice, and University Health Center - Pharmaceutical Research Center.

Cloud Computing

The HHIEA contains multiple components that are suitable to benefit from the cloud computing abstraction. Most importantly, the proposed edge server infrastructure can run on a cloud infrastructure with an adequate deployment model and largely benefit from its elasticity (i.e., on-the-fly relocation of resources to replicas that experience increased system load) and low up-front cost (i.e., decrease of initial implementation barriers). Further targets for the cloud model are the installed data warehouses as well as any of the dedicated HITs (e.g., EHRs, PHRs, PMSs, etc. that are integrated).

Grid

The HHIEA provides all of the means for the execution of grid applications: grid service discovery, grid service descriptions, and a secure virtual domain with point-to-point messaging. This resolves all of the links which are related to applications requiring large amounts of computational or data storage resources, such as Local Hospital – University Supercomputer Center and University Health Center – University Supercomputer Center.

Data Warehouse

Since the proposed infrastructure effectively supports reporting, as well as the transfer of arbitrary healthcare data, data warehouses can be installed at all of the HIT sources which require the collection and analysis of large data sets, such as Stage Agency and Pharmaceutical Research Center. Note that the creation of a data warehouse can occur through incremental event-driven reporting via publish/subscribe or through mass extraction of data from data suppliers (HIT sources) of the HIE via an adequate service. Access of the data warehouses can be either limited to the governing HIT source or made available to the whole HIE via service.

FHIR

The FHIR standard is a promising tool for the exchange of healthcare data in HITs. FHIR's focus on modern API design and ease of implementation makes it an excellent choice for usage at relevant locations in the HHIEA that need to exchange healthcare records. Therefore, the links Community Practice – Virtual Chart, Local Hospital – Virtual Chart, and PHR – Virtual Chart are strong candidates to be FHIR enabled. The same applies for all of the links that serve for gathering and aggregating medical records and for event driven reporting of medical cases. Examples for this category are the links pointing towards the State Agency and the Pharmaceutical Research Center. Depending on detailed requirements (levels of authorization, anonymization, etc.), implementations can serve FHIR resources in XML or JSON format either as payload messages of the SOA infrastructure and/or provide immediate access over adequate RESTful APIs.

FUTURE TRENDS

This section identifies and reviews trends that will impact the evolution of HIE technologies and solution approaches. First, the trend for modularization of healthcare applications is presented, as there are technologies that are nearing fruition towards a viable solution for HIE. Second, the need for integration genomic testing and research data with clinical care EHRs is discussed, which is more speculative in nature and represents an emerging need as genetic testing is utilized by medical providers as part of the diagnosis and treatment processes. Third, efforts in leveraging and extending FHIR to support healthcare systems reviewed in the first two future trends are explored.

Modular Application Architectures

Recently there have been noteworthy efforts in developing modular frameworks for healthcare applications that are coming to the forefront as possible solutions for HIE. The Substitutable Medical Applications, Reusable Technologies (SMART, 2015) platform chooses an app-centric approach to modularization, which is inspired by the use of apps on smartphone platforms. The approach is based on the abstract specification of a SMART container that encapsulates healthcare data and exposes a well-defined application programming interface (API) that allows interaction with the data. In theory, any HIT source containing relevant data (e.g., EHRs, PMSs, PHRs, etc.) can be turned into a SMART container by implementing the container API on top of the used HITs. The API is designed to be used by SMART apps, which perform tasks based on the containers data (e.g., detection of drug-drug interactions or the visualization of genomic test results). The apps are reusable in the sense that they only conform to the container API and are unaware of the details of the underlying HIT source and therefore can be executed by any container. They are substitutable, since the container abstraction decouples functionality from data and, thus, allows competition between different apps (e.g., there can be multiple apps for visualizing blood glucose levels). This approach lowers the complexity for software developers and fosters the creation of a quickly growing app collection (Mandl et al., 2012). From the perspective of HIE between multiple containers, the SMART architecture has to rely on functionalities implemented in its apps and will eventually require another level abstraction/ synchronization (ensuring that container A and container B run compatible apps for a given task) to establish collaboration. In the SMART world, each EHR vendor could be asked to provide a SMART app to their product, where the apps would all have identical APIs in terms of the services. This would allow a developer to access multiple HITs through a similar interface thereby supporting both reuse and HIE.

Another modular approach for the healthcare domain is Open mHealth (Open mHealth 2015), which provides an architecture description based on abstract components that can be used to build healthcare applications. These components are classified as data visualization units (DVU), data processing units (DPU), and cache units (CU). By providing specifications for implementing the external behavior of the components, Open mHealth aims to define the interaction of the components. This abstraction allows for separately developing components, such as, a DVU for presenting bar charts of a fitness marker can be developed separately from the DPU aggregating the data a fitness tracking service and a PHR. As a result, these components can then be combined, such as a DVU presenting the fitness marker results as a spread sheet can replace the bar chart DVU. Here the collaboration of multiple HIT sources that are built according to the Open mHealth architecture relies on the capability of expressing HIE requirements within the specifications.

Integration of Genomics into EHRs

With the progress of genomic analysis methods and the decrease of cost for genotyping procedures, genomic data and knowledge is becoming increasingly comprehensive and available. As a result, its usage for patient diagnosis, treatment, and even medication selection, may now be based on genetic information. In support of this, physicians must be brought into a position to effectively utilize genetic information for patient care. In fact, there needs to be an HIE of genetic data with the diagnostic data found in an EHR. The major initiative of note in this area is the Electronic Health Records and Genomics (eMERGE, 2007) Network that is striving towards the goal of providing approaches for combining and reconciling data from specialized DNA repositories with data collected in EHRs. This integration of research data and clinical data allows for advancing genetic research on the one side and significantly improving clinical care on the other side.

Genomic research investigations can benefit from this linkage by exploring a broad data set that is already available as a result of providing clinical care to patients (i.e., there are no additional costs for data collection) and contains information required for phenotype analysis (Denny, 2012). This facilitates studies that attempt to link certain genes to diseases or drug intolerances. Physicians equipped with EHRs that integrate genetic profiles have the means to provide care based on this additional dimension of knowledge. This includes options such as: better risk assessment and early prevention if the genetic profile of a patient indicates traits that research linked to diseases (Chute et al., 2013); improved diagnosis through decision support systems that make use of genetic profiles (Overby et al., 2013); and, accurate and personalized treatment based on results from pharamacogenomic research (Relling & Klein, 2011).

Integration of genomics data into EHRs creates a new set of requirements for HIE architectures. First, data stored in the clinical care records has to be made available for data mining in the genomics context. This includes safety and privacy preserving access to the data as well as preparation of the data by establishing common formats and extraction procedures (e.g., extraction of data from free text notes) of data relevant for genomic research. Second, EHRs have to be redesigned to match genomics enabled workflows (e.g., preemptive ordering of tests based on a patient's health history and health markers) and to process genetic test results (frequently large and highly complex data sets); this may require sophisticated user interfaces for medical providers accompanied by education and training.

Impact of FHIR on HIT and Genomics

There are numerous emerging efforts in leveraging and extending FHIR to support healthcare systems with a review of the rapidly growing research and development efforts in FHIR that include: SMART on FHIR, SMART on FHIR Genomics, and HEART profile for FHIR. The first effort, the Substitutable Medical Apps and Reusable Technology (SMART) project, was initiated by Harvard Medical School and Boston Children's Hospital with an aim to enable interoperability between medical applications by providing a specification to enable developers in the health informatics community to create medical applications once and deploy them across different HIT systems without rewriting the application code for each HIT systems (Mandl et al., 2012). SMART on FHIR (Mandel, Kreda, Mandl, Kohane, & Ramoni, 2016), is a recently released version of SMART that adopts numerous FHIR main features. This includes: FHIR data models, data formats, and API; authorization using OAuth2 (OAuth, 2016); authentication utilizing OpenID connect (OpenID, 2016); SMART profiles that integrate with FHIR profiles; and EHR user interface integration. Specifically, the SMART on FHIR reference platform has been implemented with three main servers: an API server that provides create, read, update, and delete services for all FHIR resources with an implementation of the FHIR search service; an authorization server which is a modified implementation of an open source OAuth2 and OpenID servers; and, an applications server that uses an EHR-like framework for developers to retrieve a list of patient data. By providing this reference platform, SMART on FHIR enables flexibility and innovations and enables systems to grow quickly as user needs change.

SMART on FHIR Genomics (Alterovitz, et al., 2015) is a specification that adds genomic capabilities to FHIR with the intent to integrate genomic and clinical data. The work proposes three new FHIR resource and extension definitions: a Sequence resource for capturing a patient's genetic data; a SequencingLab extension to capture the specific sequencing technique which is utilized to generate sequences; and a GeneticObservation extension to associate a phenotype to variant data. SMART on FHIR Genomics extends the SMART on FHIR platform by adding features that enable developers to bridge between the genomics and clinical communities via one integrated platform, thereby supporting the combination of different sources of genomic and electronic health record (EHR) clinical data. The end result of such a combination is the ability to develop new types of medical and healthcare applications that can be utilized for precision and personalized medicine.

The HEART Working Group (Health Relationship Trust Working Group, 2016) was formed to develop a unified set of privacy and security specifications that would be able to control authorization to RESTful APIs. As part of this effort, a HEART profile is being proposed that is capable of interacting with various authentication protocols and tools: OAuth 2.0, OpenID Connect, FHIR OAuth 2.0 Scopes, and User-Managed Access. The addition of the OAuth 2.0 protocol to the FHIR standard to prevent privacy and security issues that a FHIR implementation may face would be an important extension to FHIR to further enhance the interoperability of FHIR. The intent is to allow customized access to a set of RESTful health-related data sharing APIs that would be capable of controlling access to different portions of the API on a user/role and/or application basis. This extends OAuth 2.0 that had typically focused on the access of a client to a system to a more fine grained security access to control who can utilize which services of an API. To achieve this, the HEART profile for FHIR introduces the concept of scopes to restrict access to different parts of an API. For example, scopes can be utilized to restrict: the type of resource (e.g., Patient, Observation, etc.) to be protected; the type of access to a requested resource (e.g. read, create, and delete) which is essentially the CRUD services that can be invoked; and, the exact part of a resource to be accessed (e.g., user ID and resource ID). A scope value is a composite text that contains: the type of permission, the type of resource, and the type of access to that resource being requested.

CONCLUSION AND ONGOING EFFORTS

This paper has studied the alternative approaches to data and system integration in the healthcare domain leveraging established software architectural styles in conjunction with the emergent HL7 standard Fast Healthcare Interoperability Resources (FHIR). Towards this goal, the concept of health information exchange (HIE) has been introduced as a means to integrate data from multiple HIT sources (e.g., EHRs, PHRs, PMSs) which required the reconciliation of a set of diverse requirements to HIE-enabled systems and a review of the challenges that HIE architectures need to overcome in order to establish interactions between these systems that facilitate the collaboration of medical providers. In support of this goal, a selection of software architectures (service-oriented architecture, grid computing, publish/subscribe paradigm, data warehousing, cloud computing, and FHIR) was studied, presented and discussed in the context of these requirements, which allowed us to highlight strengths and weaknesses that each architecture exhibits for a given aspect of HIE. To allow a further inspection of the capabilities of the different introduced architectural styles, a regional HIE scenario was detailed around a real-life HIT infrastructure that connects multiple and diverse healthcare stakeholders and is based on significant input from our medical collaborator. From this scenario, a set of collaborative links was extracted that can be used as a benchmark for the capabilities of an HIE architecture. Utilizing the architectures and the scenario as a basis, elements from SOA, grid, publish/subscribe, data warehouse architectures, cloud computing, FHIR, and a replication approach to data storage were subsequently utilized to propose a hybrid HIE architecture (HHIEA), which has been shown to meet the established requirements and the identified links. The usage of the studied architectural styles in conjunction with FHIR for the proposed HHIEA is coupled with a focus on service-oriented components as attained using FHIR which can ease the process of integrating new HIT sources over time. This paper was concluded through a discussion of modular application architectures for healthcare that promote both abstract application development and HIE, coupled with the emerging need for integrating genetic and patient data for more effective treatment.

With FHIR currently affecting many existing approaches to HIE, such as the aforementioned SMART platform (Mandel et al., 2016), we are currently revisiting the exemplary topic of patient-driven medication reconciliation (Ziminski et al., 2012) in three areas. Firstly, we seek to analyze the impact of FHIR and its usage of industry standard technologies on software implementation complexity in comparison to previously chosen concepts of modeling healthcare data as well as to determine possible requirements for improvement. Secondly, with today's particular focus on mHealth applications, as discussed in this paper, we aim to qualify the capabilities of FHIR to make existing platforms mobile-friendly together with identifying any expected conceptual roadblocks. Finally, our most ambitious effort is to design and implement a portion of HHIEA as given in Figure 18 as a proof-of-concept prototype to demonstrate the ideas presented in this paper.

REFERENCES

Aitken, M. (2013). Patient apps for improved healthcare: from novelty to mainstream. Retrieved from http://www.imshealth.com/en/thought-leadership/ims-institute/reports/patient-apps-for-improved-healthcare/

Alterovitz, G., Warner, J., Zhang, P., Chen, Y., Ullman-Cullere, M., Kreda, D., & Kohane, I. S. (2015). SMART on FHIR Genomics: Facilitating standardized clinico-genomic apps. *Journal of the American Medical Informatics Association*. PMID:26198304

Andrews, C., & Mack, R. (2011). IBM to Collaborate with Nuance to Apply IBM's "Watson" Analytics Technology to Healthcare. *IBM News Room.* Retrieved from http://www-03.ibm.com/press/us/en/pressrelease/33726.wss

Apache, ESB (2008). *Apache Software Foundation Synapse ESB*. Retrieved from http://synapse.apache.org

Apple Health App (2015). *Apple Health App.* Retrieved from https://www.apple.com/ios/ios8/health/

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., & Zaharia, M. et al. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50–58. doi:10.1145/1721654.1721672

Baihan, M., & Demurjian, S. (in press). A Framework for Secure and Interoperable Cloud Computing". Submitted to *Springer Research Advances in Cloud Computing.* Retrieved from http://www.cloudbus.org/racc/

Baihan, M., Rivera Sánchez, K. Y., Shao, X., Gilman, C., Demurjian, S., & Agresta, T. (in press). A Blueprint for Designing and Developing an mHealth Application for Diverse Stakeholders Utilizing Fast Healthcare Interoperability Resources. *Contemporary Applications of Mobile Computing in Healthcare Settings*. Retrieved from http://www.igi-global.com/publish/call-for-papers/call-details/2287

Bell, D. E., & LaPadula, L. J. (2003). *Secure computer systems: Mathematical foundations*. In I. Bilykh, Y. Bychkov, D. Dahlem et al. (Eds.), Can GRID services provide answers to the challenges of national health information sharing? *Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research* (pp. 39-53). IBM Press.

BizTalk Server. (2006). *Microsoft BizTalk Server*. Retrieved from http://www.microsoft.com/biztalk/

Blue Button. (2013). *Blue Button*. Retrieved from http://www.healthit.gov/patients-families/blue-button/

caBIG. (2004). *The Cancer Biomedical Informatics Grid (caBIG)*. Retrieved from https://cabig.nci.nih.gov/

CCR. (2012). *Continuity of Care Record (CCR)*. Retrieved from http://www.aafp.org/practice-management/health-it/astm.html

Chute, C. G., Ullman-Cullere, M., Wood, G. M., Lin, S. M., He, M., & Pathak, J. (2013). Some experiences and opportunities for big data in translational research. *Genetics in Medicine*, *15*(10), 802–809. doi:10.1038/gim.2013.121 PMID:24008998

CMS. (2013). *Centers of Medicare and Medicaid Electronic Health Records Incentive Programs.* Retrieved from www.cms.gov/EHRIncentivePrograms

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., & Weerawarana, S. (2002). Unraveling the Web services web: An introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing*, *6*(2), 86–93. doi:10.1109/4236.991449

De La Rosa Algarín, A., Demurjian, S. A., Berhe, S., & Pavlich-Mariscal, J. A. (2012). A security framework for XML schemas and documents for healthcare. *Bioinformatics and Biomedicine Workshops (BIBMW), 2012 IEEE International Conference on* (pp. 782–789). IEEE.

De La Rosa Algarín, A., Ziminski, T. B., Demurjian, S. A., Kuykendall, R., & Rivera Sánchez, Y. (2013). Defining and Enforcing XACML Role-Based Security Policies within an XML Security Framework. *Proceedings of 9th International Conference on Web Information Systems and Technologies (WEBIST 2013)* (pp. 16–25). doi:10.5220/0004366200160025

Demurjian, S. A., Saripalle, R., & Berhe, S. (2009). An Integrated Ontology Framework for Health Information Exchange. *SEKE*, *09*, 575–580.

Denny, J. C. (2012). Mining electronic health records in the genomics era. *PLoS Computational Biology*, *8*(12), e1002823. doi:10.1371/journal.pcbi.1002823 PMID:23300414

eMERGE. (2007). *Electronic Medical Records and Genomics (eMERGE) Network*. Retrieved from http://emerge.mc.vanderbilt.edu/

Enrado, P. (2011). Why shuttered RHIO CareSpark's chairman is not giving up. *Government Health IT*. Retrieved from http://www.govhealthit.com/news/why-shuttered-rhio-caresparks-chairman-not-giving

Eugster, P. T., Felber, P. A., Guerraoui, R., & Kermarrec, A.-M. (2003). The many faces of publish/subscribe. *ACM Computing Surveys*, *35*(2), 114–131. doi:10.1145/857076.857078

FERPA. (1974). *Family Educational Rights and Privacy Act (FERPA)*. Retrieved from http://www.ed.gov/policy/gen/guid/fpco/ferpa/

Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security*, *4*(3), 224–274. doi:10.1145/501978.501980

Foster, I. (2002). What is the grid? a three point checklist. *GRID today, 1*(6).

Gomes, A. T. A., Ziviani, A., Correa, B. S. P. M., Teixeira, I. M., & Moreira, V. M. (2012). SPLiCE: a software product line for healthcare. *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium* (pp. 721–726). ACM. doi:10.1145/2110363.2110447

Google Fit. (2015). *Google Fit.* Retrieved from https://developers.google.com/fit/

HL7. (2016) *Health Level Seven International*. Retrieved from http://www.hl7.org/

HL7 V2. (2016). *Health Level 7 Version 2*. Retrieved from http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185

HL7 V3. (2016). *Health Level Version 3*. Retrieved from https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186

HL7 CDA. (2007). *HL7 Clinical Document Architecture (CDA)*. Retrieved from http://www.hl7.org/implement/standards/

HL7 FHIR. (2016). *FHIR Overview.* Retrieved from http://hl7.org/fhir/overview.html

Haas, H., & Brown, A. (2004). *Web Services Glossary*. World Wide Web Consortium (W3C). Retrieved from http://www.w3.org/TR/ws-gloss/

Health Relationship Trust Working Group. (2016). *HEART profile for FHIR*. Retrieved from http://openid.net/wg/heart/

HIMSS. (2015). *Meaningful use stage 3 final rule.* Retrieved from http://www.himss.org/ResourceLibrary/genResourceDetailPDF.aspx?ItemNumber=4498

HIPAA. (1996). *Health Insurance Portability and Accountability Act (HIPAA)*. Retrieved from http://www.hhs.gov/ocr/privacy/

i2b2. (2004). *Informatics for Integrating Biology & the Bedside (i2b2).* Retrieved from https://www.i2b2.org/

Inmon, W. H. (2005). *Building the Data Warehouse* (4th ed.). New York, NY, USA: John Wiley & Sons, Inc.

Kenny, P., Parsons, T., Gratch, J., & Rizzo, A. (2008). Virtual humans for assisted health care. *Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments* (pp. 1–4). ACM.

Logicworks. (2015). *Logicworks Healthcare Solutions*. Retrieved from http://www.logicworks.net/healthcare-cloud-solutions

Mandel, J. C., Kreda, D. A., Mandl, K. D., Kohane, I. S., & Ramoni, R. B. (2016). SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. *Journal of the American Medical Informatics Association*. Retrieved from http://docs.smarthealthit.org/

Mandl, K. D., Mandel, J. C., Murphy, S. N., Bernstam, E. V., Ramoni, R. L., Kreda, D. A., & Kohane, I. S. et al. (2012). The SMART Platform: Early experience enabling substitutable applications for electronic health records. *Journal of the American Medical Informatics Association*, *19*(4), 597–603. doi:10.1136/amiajnl-2011-000622 PMID:22427539

Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing (Draft) Recommendations of the National Institute of Standards and Technology. *NIST*. Retrieved from http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc

Microsoft Cloud Services for Health. (2015). *Microsoft Cloud Services for Health*. Retrieved from http://www.microsoft.com/health/en-ca/initiatives/Pages/cloud-services-for-health.aspx

Microsoft HealthVault. (2007). *Microsoft HealthVault Personal Health Record*. Retrieved from https://www.healthvault.com/

Na, S., & Cheon, S. (2000). Role delegation in role-based access control. *Proceedings of the fifth ACM workshop on Role-based access control* (pp. 39–44). ACM. doi:10.1145/344287.344300

OAuth. (2016). *About OAuth 2.0.* Retrieved from https://oauth.net/2/

ONC. (2015). *Office of the National Coordinator for Health Information Technology Product List.* Retrieved from http://oncchpl.force.com/ehrcert

Open, I. D. (2016). *About OpenID Connect.* Retrieved from http://openid.net/connect/

Open mHealth. (2015). *Open mHealth.* Retrieved from http://openmhealth.org/

OpenEMR. (2012). *OpenEMR electronic health record*. Retrieved from http://www.open-emr.org/

OpenESB. (2012). *Sun Microsystems OpenESB*. Retrieved from http://open-esb.dev.java.net/

OpenMRS. (2004). *OpenMRS electronic medical record system platform.* Retrieved from http://openmrs.org/

Oracle, ESB (2012). *Oracle ESB*. Retrieved from http://www.oracle.com/appserver/esb.html

Overby, C. L., Kohane, I., Kannry, J. L., Williams, M. S., Starren, J., Bottinger, E., & Hripcsak, G. et al. (2013). Opportunities for genomic clinical decision support interventions. *Genetics in Medicine*, *15*(10), 817–823. doi:10.1038/gim.2013.128 PMID:24051479

Practice Fusion EHR. (2015). *Practice Fusion EHR*. Retrieved from http://www.practicefusion.com/electronic-health-record-ehr/

President's Council of Advisors on Science and Technology. (2010). *Realizing the Full Potential of Health Information Technology to Improve Healthcare for Americans: The Path Forward.* Retrieved from http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast-health-it-report.pdf

Prokosch, H., & Ganslandt, T. (2009). Perspectives for medical informatics. Reusing the electronic medical record for clinical research. *Methods of Information in Medicine*, *48*(1), 38–44. PMID:19151882

Relling, M., & Klein, T. (2011). CPIC: Clinical pharmacogenetics implementation consortium of the pharmacogenomics research network. *Clinical Pharmacology and Therapeutics*, *89*(3), 464–467. doi:10.1038/clpt.2010.279 PMID:21270786

ResearchKit. (2015). *ResearchKit.* Retrieved from https://www.apple.com/researchkit/

Rivera Sánchez, Y. K., Demurjian, S., Conover, J., Agresta, T., Shao, X., & Diamond, M. (2016). An Approach for Role-Based Access Control in Mobile Applications. In S. Mukherja (Ed.), Mobile Application Development, Usability, and Security. Hershey, PA: IGI Global.

Robinson, B. (2010). CalRHIO shuts down after missing out on HIE bid. *Government Health IT*. Retrieved from http://www.govhealthit.com/news/calrhio-shuts-down-after-missing-out-hie-bid

Rosen, M. (2008). *Applied SOA: service-oriented architecture and design strategies*. Wiley.

Ryan, A., & Eklund, P. (2008). A framework for semantic interoperability in healthcare: A service oriented architecture based on health informatics standards. *Studies in Health Technology and Informatics*, *136*, 759. PMID:18487823

Sanzi, E., Demurjian, S., Agresta, T., & Murphy, A. (2016). Trust Profiling to Enable Adaptive Trust Negotiation in Mobile Devices. In S. Mukherja (Ed.), Mobile Application Development, Usability, and Security. Hershey, PA: IGI Global.

SHARP. (2013). *Strategic Health IT Advanced Research Projects (SHARP)*. Retrieved from http://www.healthit.gov/policy-researchers-implementers/strategic-health-it-advanced-research-projects-sharp

Shortliffe, E. H., & Cimino, J. J. (2006). *Biomedical informatics: computer applications in health care and biomedicine* (pp. 46–79). Springer Verlag. doi:10.1007/0-387-36278-9_2

Singh, J., Vargas, L., Bacon, J., & Moody, K. (2008). Policy-based information sharing in publish/subscribe middleware. *Proceedings of the IEEE Workshop on Policies for Distributed Systems and Networks POLICY '08* (pp. 137–144). IEEE.

SMART. (2015). *Substitutable Medical Apps & Reusable Technology (SMART)*. Retrieved from http://smartplatforms.org/

UCSF. (2016). *Mobile apps and resources*. Retrieved from http://guides.ucsf.edu/c.php?g=100993&p=654826

University Health Network. (2016). *HAPI-FHIR*. Retrieved from http://hapifhir.io

VistA (2003). *Veterans Health Information Systems and Technology Architecture* (*VistA) health record and information system*. Retrieved from http://www.worldvista.org/

VMWare vCloud. (2015). *VMWare vCloud for Healthcare*. Retrieved from http://www.vmware.com/industry/healthcare/

W3C. (2008). *World Wide Web Consortium (W3C) tutorial site*. Retrieved from http://www.w3schools.com/

Wager, K. A., Lee, F. W., & Glaser, J. P. (2009). *Health care information systems: a practical approach for health care management*. John Wiley and Sons.

Walker, J., Pan, E., Johnston, D., & Adler-Milstein, J. (2005). The value of health care information exchange and interoperability. *Health Affairs*, 24. PMID:15659453

WebSphere ESB. (2008). *IBM WebSphere Enterprise Service Bus*. Retrieved from http://www.ibm.com/software/integration/wsesb/

World Community Grid. (2004). *World Community Grid*. Retrieved from http://www.worldcommunitygrid.org/

XACML. (2003). *Introduction to XACML*. Retrieved from https://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html

Zeh, T. (2003). Data Warehousing als Organisationskonzept des Datenmanagements. *Informatik-Forschung und Entwicklung*, *18*(1), 32–38. doi:10.1007/s00450-003-0130-8

Ziminski, T. B., De la Rosa Algarín, A., Saripalle, R., Demurjian, S. A., & Jackson, E. (2012). SMARTSync: Towards Patient-Driven Medication Reconciliation Using the SMART *Framework. Proceedings of 2012 International Workshop on Biomedical and Health Informatics (BHI 2012)* (pp. 806-813)

Ziminski, T. B., Demurjian, S. A., Sanzi, E., & Agresta, T. (2016). Toward integrating healthcare data and systems: A study of architectural alternatives. In Maximizing healthcare delivery and management through technology integration (pp. 270–304). Hershey, PA: IGI Global. doi:10.4018/978-1-4666-9446-0.ch016

KEY TERMS AND DEFINITIONS

Health Information Exchange (HIE): The electronic transfer of healthcare data among distinct healthcare organizations and their HIT systems. HIE makes medical and health data available to healthcare stakeholders and enables collaboration.

Health Information Technology System (HIT): A system that captures, stores, processes, or shares information about the health of individuals or supports processes and workflows of healthcare domain stakeholders.

Architectural Alternative: A software architecture style that is suitable for a given task (e.g., data integration) under a set of domain specific requirements.

Hybrid HIE Architecture (HHIEA): An HIE-specific system architecture incorporating multiple architectural alternatives.

Collaborative Link: A relationship between two domain stakeholders, in which collaboration requires the integration of systems or data.

Service-oriented Architecture (SOA): An architectural style based on loosely coupled software services that collaborate through a connecting framework.

Data Warehouse: An architectural style that collects data from multiple sources to provide a uniform view for querying, analysis, and decision making tasks.

Publish/Subscribe Architecture: An architectural style describing the asynchronous message passing between publishers, subscribers, and optional brokers.

Cloud Computing: An architectural style based on the provision of abstract computing resources, which are capable to dynamically grow and shrink depending on changing requirements.

Fast Healthcare Interoperability Resources (FHIR): A standard for expressing healthcare data as so called “resources” and for the exchange of said resources over an application programming interface (API).