P1 P2 P3 P4 P5 P6 P7 P8

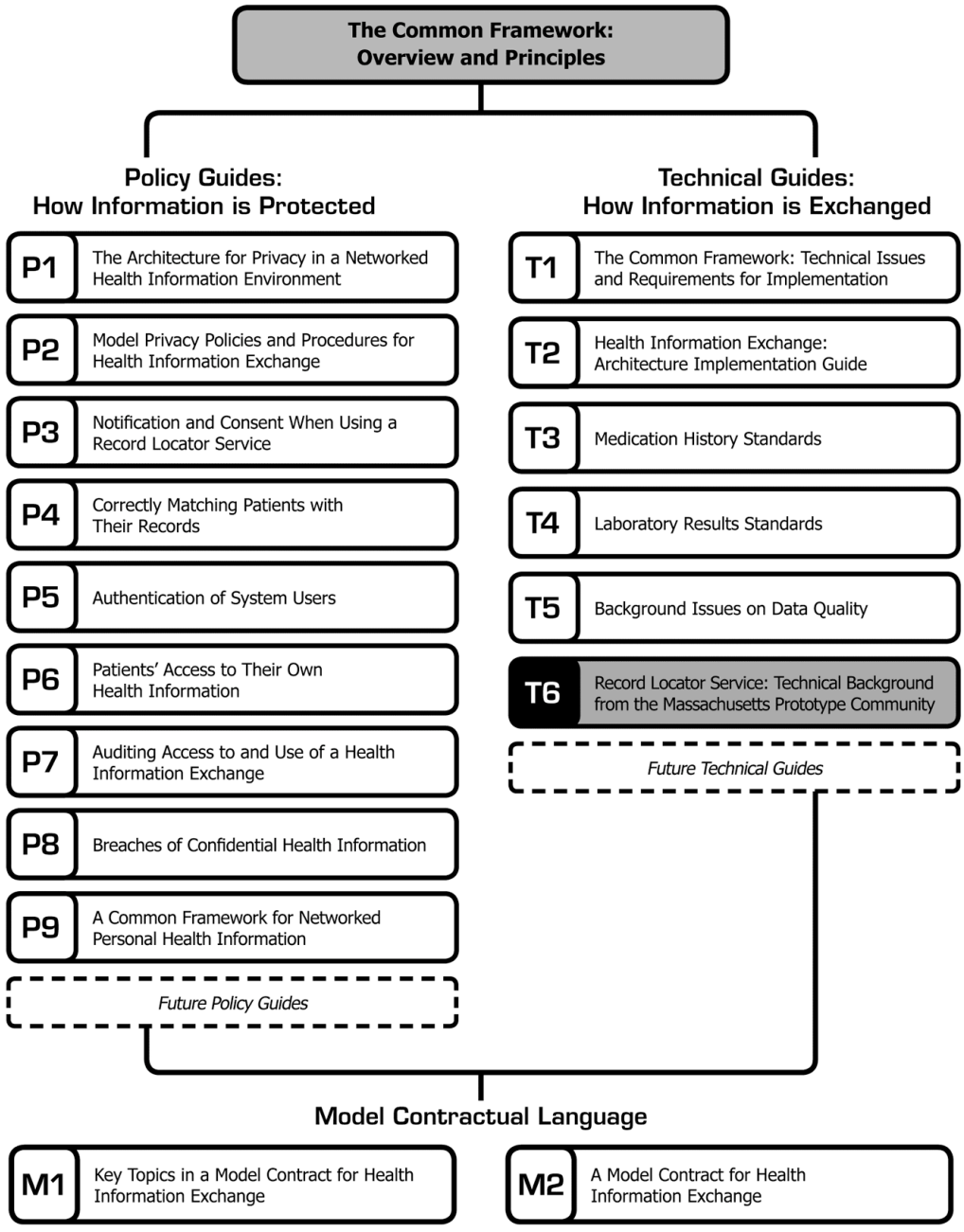
T1 T2 T3 T4 T5 T6 M1 M2

Record Locator Service:

Technical Background from the Massachusetts Prototype Community

Record Locator Service – Technical Background from the Massachusetts Prototype Community

The document you are reading is part of *The* ***Connecting for Health*** *Common Framework,* which is available in full and in its most current version at: http://www.connectingforhealth.org/. The Common Framework will be revised and expanded over time. As of October 2006, the Common Framework included the following published components:



Record Locator Service – Technical Background from the Massachusetts Prototype Community*

This document describes the early design process for the Record Locator Service (RLS) as implemented in Massachusetts, and is included here as background on the technical conversation around the design of the **Connecting for Health** prototype. It is included here as a background guide to the issues surrounding the design of the RLS as constructed in Massachusetts; as noted in "The Common Framework: Technical Issues and Requirements for Implementation," the placement of aggregation services can vary between sub-network organizations (SNOs). In this document, aggregation takes place via a clinical data exchange service; other architectural models are possible.

In addition to the overview of the architectural design decisions included in "The Common Framework: Technical Issues and Requirements for Implementation," the technical details surrounding message exchange in the current prototype are documented in the "Health Information Exchange: Architecture Implementation Guide."

* **Connecting for Health** thanks Computer Sciences Corporation (CSC) for drafting this paper.

Revision History

Date	Version	Description	Author
2005-02-18	0.1	▪ Initial (strawman) version where architecture discussions and design to date are documented for further review	CSC
2005-03-01	0.2	▪ Revised draft based on feedback from internal review. Added content on security.	CSC
2005-03-23	0.3	▪ Revised based on feedback from reviewers of 0.2 ▪ Changed conceptual application architecture to explicitly support peer-to-peer messaging and support inter-RLS messaging ▪ Moved revised conceptual application architecture diagram from Implementation view to Logical view ▪ Moved process diagrams from logical to process view. Added content on message exchange patterns to process specifications	CSC
2005-03-25	0.4	▪ Revised content based on internal review. ▪ Packaged to publish to Markle Connecting for Health Technical Subcommittee ▪ Retitled document: Framework Technical Overview	CSC
2005-04-12	0.5	▪ Incorporated feedback from review of 0.4 by: Markle Connecting for Health Technical Subcommittee and MA-SHARE Technical Advisory Board. Comments provided in Appendix with responses. ▪ Changed architecture diagram to depict RLS and CDX Gateway as two solutions that the RLS Prototype project will develop to be flexible collection of a loosely-coupled services. Shows separation of RLS and other components more clearly. Section 5.2 ▪ Added sequence diagrams to process view (Section 6) indicating processing logic for transactions supported by RLS	CSC
2005-04-15	0.6	▪ Incorporated comments from internal review ▪ Added discussion items 4.5 (patient lookup with local MRN) and 4.6 (query-time matching)	CSC
2005-05-20	1.0	▪ Language edits: cleaned up references to RLS Exchange ▪ Changed Figures: 3 and 4 to align with more precise definition of RLS scope (vis-à-vis clinical data exchange) ▪ Modified Figure 13 and added Figure 14 to provide more details of security architecture and process	CSC
2005-11-22	1.1	▪ Removed implementation specifications pertinent to only Massachusetts pilot. Reoriented content to be more generic. ▪ Removed discussion items that were relevant to prototype architecture decisions.	CSC

Table of Contents

1 INTRODUCTION 7
1.1 PURPOSE 9
1.2 SCOPE 9
1.3 REFERENCES 9
1.4 DOCUMENT OVERVIEW 10
1.5 ARCHITECTURAL REPRESENTATION 11

2 ARCHITECTURAL GOALS, PRINCIPLES AND CONSTRAINTS 13
2.1 GOALS 13
2.2 PRINCIPLES 13

3 USE-CASE VIEW 17
3.1 RLS FUNCTIONS 17
3.2 USE CASES 18
3.3 USE-CASE REALIZATIONS 21
3.4 SECURITY, PATIENT PRIVACY AND CONSENT MANAGEMENT 22
3.4.1 Identity management 23
3.4.2 Confidentiality, Authentication, Integrity & Non-repudiation 23
3.4.3 Patient Data Privacy 23
3.4.4 Consent Management 24
3.5 PATIENTS RECORDS LINKING AND MATCHING 24

4 LOGICAL VIEW 26
4.1 CONCEPTUAL RLS-SERVICES VIEW 26
4.2 RLS APPLICATION SERVICES 28
4.3 GATEWAY SERVICES 30
4.4 RLS-BASED NETWORKS 33
4.5 REGIONAL AND NATIONAL NETWORK SUPPORT 35

5 PROCESS VIEW 36
5.1 PATIENT LOOKUP AND PEER TO PEER MEDICAL RECORDS RETRIEVAL 36
5.2 PATIENT INDEX PUBLISH 37
5.3 CENTRALLY MEDIATED MEDICAL RECORDS RETRIEVAL 37
5.4 CENTRAL MEDICAL RECORDS AGGREGATION 38
5.5 SECURITY PROCESSES 39
5.6 MESSAGING PATTERNS 41

6 IMPLEMENTATION VIEW 46
6.1 OVERVIEW 46
6.2 COMPONENTS AND LAYERS 47
6.3 IMPLEMENTATION TOPOLOGY OPTIONS 52
6.4 SECURITY MODEL 54
6.5 IMPLEMENTATION PLATFORMS 55
6.6 INTERCONNECTIVITY AND DATA STANDARDS 56
6.6.1 Messaging and Transport Standards 57
6.6.2 Domain Data Standards 60
6.6.3 Comprehensive Standards List 61

7 DEPLOYMENT VIEW 64
7.1 SERVICES MANAGEMENT 66
7.2 SECURITY SERVICES 66

8 DATA VIEW 67
8.1 CMPI INFORMATION MODEL 67
8.2 LOGICAL DATA MODEL 68
8.2.1 Identifier Attributes 70
8.3 PHYSICAL DATA MODEL 72
8.3.1 Data Quality Management 74

8.3.2 Data Cache *74*

9 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS **75**

List of Figures

Figure 1: Architecture views and their contents 12
Figure 2: RLS Long term Concept of Operations 18
Figure 3: RLS Prototype Use Case Diagram 19
Figure 4 Lookup Patient and Publish Patient Index Activity Diagrams 22
Figure 5 RLS Conceptual Architecture of Operation 26
Figure 6 Service Oriented Interoperation 27
Figure 7: RLS Distribution of Components 31
Figure 8: Gateway based interaction in a health information network 33
Figure 9 Network of clinical systems communicating peer-to-peer 34
Figure 10 Network of RLS-based networks (potentially used by RHIO) 35
Figure 11: Patient Lookup with RLS and Medical Records Retrieval through CDX Gateways 36
Figure 12 Patient Publish into RLS Patient Index 37
Figure 13 Centrally Mediated Patient Lookup and Record Retrieval (hosted gateway) 38
Figure 14 Patient Lookup and Records Retrieval -- In One Step 38
Figure 15 RLS authentication service 40
Figure 16: Authentication Mechanisms for Patient Lookup and Medical Records Retrieval 41
Figure 17 Patient lookup sequence diagram 44
Figure 18 Publish patient index sequence diagram 45
Figure 19 RLS and CDX Gateway components and sample (prototype) implementation platforms 48
Figure 20 Web services stack 58
Figure 21 WS-I Basic Profile Web Services stack 59
Figure 22 Interoperability Network Layers 61
Figure 23 Potential Production Deployment View of RLS 65
Figure 24 Information Model View 67
Figure 25 Logical Data Model 69
Figure 26 Patient Identifier Composition in CMPI 73

List of Tables

Table 1 List of Architecturally Significant Use Cases 20
Table 2 List of Messaging Interactions supported by RLS Prototype 42
Table 3 CDX Gateway Service / Components Description 49
Table 4 RLS Components Description 51
Table 6 Prototype Platform and Options 55

1 Introduction

The Record Locator Service (RLS) is envisioned as the key infrastructure component of the 'Common Framework', which Markle Foundation Connecting for Health (CfH)[†] has proposed to facilitate healthcare information networks in the USA. The common framework is a set of standards, policies, and methodologies intended to ensure secure and reliable connectivity between healthcare systems and enterprises.

The common framework includes the essential set of standards and policies that would allow healthcare information networks to interoperate with each other. This would enable communities and regional networks to connect and incrementally grow into a national healthcare information network, as a "network of networks".

Building a national network through internetworking multiple regional and local health information networks implies a natural bias towards decentralization. A centralized national patient registry or clinical data repository is not considered a realistic objective. Adoption of common architecture and protocols across the networks and sub-networks would, similarly, suggest decentralization in sub-networks, with data stored in separate locations to be accessed when needed. Leaving patient data where they are now, in the healthcare enterprises' clinical data sources also provides for appropriate patient data privacy safeguards and clear accountability for medical data ownership/stewardship.

This does not preclude sub-networks based on a regional data repository or a community master patient index. As long as the networks support the principles and protocols of the common framework, they would be capable of interoperation with other networks. Smaller participants may choose to use data aggregators to expose their clinical data securely and reliably. As the CfH *Roadmap* states, "Because many providers will not be able or perhaps willing to provide the levels of service required to participate in a federation, they may have to contract with business associates (in the HIPAA sense) to store their data in a repository that will sustain these service levels."[‡]

The common framework that underlies the decentralized healthcare information network is expected to need a small set of critical technical infrastructure components to support interoperability. The RLS is one of them.

[†] http://www.connectingforhealth.org
[‡] [CfH2004]

The RLS provides authorized users of a regional health information network with pointers to the location of patient health information across the network nodes, i.e. the clinical data sources. This would enable users to access and integrate patient healthcare information from the distributed sources without national patient identifiers or centralized databases. Such an integrated view of patient clinical data would help achieve the CfH vision of improved patient safety and quality of care, and reduced costs of healthcare delivery.

Massachusetts SHARE (Simplifying Healthcare Among Regional Entities)[§], a regional collaborative initiative operated by the Massachusetts Health Data Consortium, seeks to foster improvements in community clinical connectivity, allowing appropriate sharing of inter-organizational healthcare data among the various participants in the healthcare system – including patients, doctors and other practitioners, hospitals, government, insurers, HMOs and other payers. MA-SHARE promotes the inter-organizational exchange of healthcare data using information technology, standards and administrative simplification, in order to make accurate clinical health information available wherever needed in an efficient, cost-effective and safe manner.

MA-SHARE's vision includes the goal of building a utility service that would enable member organizations to hook up to the regional healthcare network (or "grid") simply and cost-effectively. The RLS architecture advances the design of such a utility service that would connect the healthcare systems in a community securely over the Internet.

This document describes the proposed architecture of the RLS, and provides an overview of the technical components of the common framework. The RLS prototype project tests the architecture presented here and demonstrates the viability of direct peer-to-peer interoperability of disparate electronic health record (EHR) systems that are the ultimate source of clinical data in the network. This document serves as the primary record of all architectural design decisions made in the course of developing the RLS prototype.

[§] http://www.mahealthdata.org/ma-share/index.html

1.1 Purpose

The Framework Technical Overview defines and describes the basic functional components that make up the RLS, and the interfaces between these components. The document presents the RLS architecture using a number of different architectural views to depict different aspects of the system, and outlines the data and transport standards used in accessing the services offered by RLS. RLS plays a critical role in the healthcare interoperability common framework, and the RLS service architecture is aligned with the clinical data exchange processes that the common framework also supports. This document also conveys the architecturally significant trade-offs and decisions which have been made in designing the system and the network.

Business stakeholders may use this document to validate the functionality of RLS in the patient care setting, and to gain understanding of the major services provided by the system. The RLS architecture is intended to serve as a reference for system designers to guide the detailed design of the system during the elaboration and construction phases of a system development project.

1.2 Scope

The Framework Technical Overview provides details of the technical architecture of the Record Locator Service prototype and outlines the strategy to meet the architectural (longer-term) requirements for the RLS as defined in the Markle Foundation Connecting for Health Common Framework Record Locator Service Reference Implementation Statement of Work[**].

Technical architecture covers the domains of data, application and technology infrastructure. While technical architecture needs to be developed in the context of the organizational and business process architecture, these domains are out of scope of Framework Technical Overview. This document uses prior work by CfH to define the RLS business context and defines the technical architecture to fulfill the use case requirements thereof. The other important component of the common framework is the policy framework produced by the CfH Policy Sub-committee. This element of the framework was under development during the production of this technical architecture document; policy based requirements have been considered where available.

1.3 References

[CfH2004] Achieving Electronic Connectivity in Healthcare, A Preliminary Roadmap from the Nation's Public and Private Sector Healthcare Leaders, Connecting for Health, Markle Foundation, July 2004.

[**] [CfH2005a]

[CfH2004a] Accurately Linking Information for Healthcare Quality and Safety, Connecting for Health, Markle Foundation, Draft Final Report by Working Group on Accurately Linking Information for Healthcare Quality and Safety, September 2004.
[CfH2004b] Connecting for Health Reference Implementation Launch, In Collaboration with the RWJF and Foundation for eHealth Initiative Confidential Draft Document for Discussion Only October 29, 2004
[CfH2005] Linking Healthcare Information: Proposed Methods for Improving Care and Protecting Privacy, Carol Diamond, Connecting for Health, Markle Foundation, HIMSS 2005
[CfH2005a] Markle Foundation Connecting for Health Common Framework Record Locator Service Reference Implementation Statement of Work, CSC, February 2005.
[CfH2005b] Linking Healthcare Information: Proposed Methods for Improving Care and Protecting Privacy, Working Group on Accurately Linking Information for Healthcare Quality and Safety, February 2005.
[MAeHC2004a] Expanding the Use of Electronic Health Records and Establishing a Regional Health Information Infrastructure in Massachusetts, Request for Applications, December 6, 2004, The Massachusetts eHealth Collaborative.
[MHDC2004a] Analysis of Approach to Uniquely Identifying Patients in Massachusetts, Massachusetts Health Data Consortium, Inc. 2004

1.4 Document Overview

The Framework Technical Overview document provides a high level overview of the software artifacts that make up the RLS. The document lays out the key business processes that the RLS is intended to support, a logical view of the components and their behavior, and the proposed deployment of the software on completion of development. The remainder of this section defines the concept of software architecture and describes the notation used to document it.

Section 2 defines the goals, principles, and constraints of the RLS architecture.

Section 3 lists the subset of use cases and scenarios that impact the architectural design of the system. Use cases represent the major business or functional requirements that the software is expected to meet.

Section 4 shows the decomposition of the solution into a set of logical elements, i.e., classes, subsystems, packages, and collaborations.

Section 5 presents the process structure of the system. The process view maps the logical view elements to the processes and threads in the solution.

Section 6 presents the implementation view of the system, i.e. the decomposition of the system into layers and packages. Alternative implementation models are presented that may be appropriate for specific scenarios.

Section 7 presents the deployment view, which maps the prototype components to a set of hardware and network nodes on which they execute. Given that this is an architecture document

Section 8 provides a view of the persistent data storage of the system to the extent that this is significant in a network with minimal central data storage.

A glossary of key terms, abbreviations, and acronyms used in this document is provided in Section 9.

1.5 Architectural Representation

The RLS software architecture defines the overall structure of the system in terms of the behavior of its components. Software architecture needs to be viewed from multiple perspectives and at different levels of abstraction to gain a full understanding of the system. In this document, the following architectural views are used:

- Use case view: outlines the functional requirements of RLS from an end-users perspective.
- Logical view: where the major subsystems and components of RLS are identified and a conceptual view of their working provided.
- Process view: describes the runtime behavior of RLS components in meeting key functional requirements.
- Implementation view: provides the organization of the RLS software artifacts in terms of layering and packaging.
- Deployment view: describes how the various RLS software packages and runtime components are deployed on hardware and network nodes.

These views are shown in Figure 1, which illustrates the central role of the use-case (or business oriented) view as the driver of the whole software architecture. This architecture representation follows the 'Rational Unified Process' reference architecture standards[††]. The system stakeholders that are primary audiences of the views are shown in the callout boxes attached to each view.

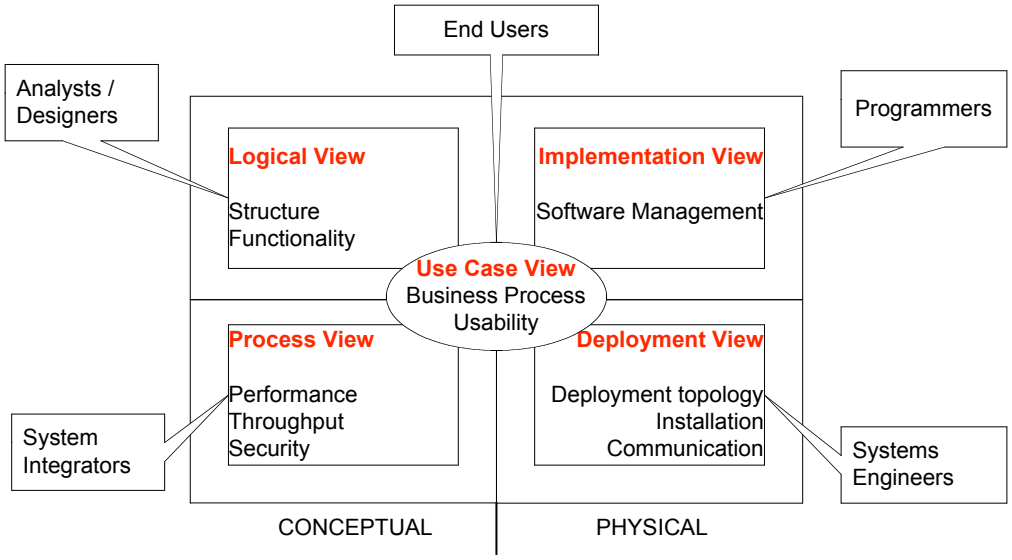


Figure 1: Architecture views and their contents

This document presents architectural views in the form of models (or diagrams) that use the Unified Modeling Language (UML) notation, where applicable.[‡‡]

The keywords "MUST", "REQUIRED", "SHALL", "SHOULD", "RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in IETF Network Working Group RFC2119[§§].

[††] Reference Architectures, The Best of Best Practices, P.R. Reed, Rational Edge, September 2002
[‡‡] Unified Modeling Language UML Resource Page, http://www.uml.org/, Object Management Group, 2005
[§§] Key words for use in RFCs to Indicate Requirement Levels, http://www.ietf.org/rfc/rfc2119.txt

2 Architectural Goals, Principles and Constraints

Architecture best practice calls for the definition of common principles at the outset to serve as a consistent basis for design decisions to be made downstream. Information Technology (IT) architectural principles define the fundamental rules and guidelines for the development and deployment of IT resources and assets. They reflect a level of consensus among the various stakeholders of the system, and form the basis for making coherent and consistent architecture and design decisions.

Architectural principles are, therefore, high level statements that govern the system architecture development process. Based on the CfH charter and the longer term MA-SHARE vision, the following RLS architecture goals, principles and constraints are proposed.

2.1 Goals

The guiding vision for RLS is that to provide directory and registry services for a regional health information network which supports the interoperability between disparate healthcare information systems in a community, reducing healthcare delivery costs and improving quality of care as well as patient safety.

CfH defines the primary objective of the Reference Implementation of the Record Locator Service as: to validate a set of standards that would, if implemented across communities, enable health information exchange within and between communities regardless of the hardware and software platforms used[***]. Such standards and profiles are part of an interoperability architectural framework, which has been referred to as the 'Common Framework' by CfH. The technical standards would align with the other components of the common framework, which includes policies and methods.

The RLS is assumed to operate under the auspices of a Regional Health Information Organization (RHIO) that coordinates the various healthcare enterprises in the region (or community). RHIOs serve as distributed hubs of a prospective National Health Information Network.

2.2 Principles

- **Patient privacy protection**: Clinical data sharing shall be subject to very stringent privacy and security constraints. All access to patient health data must be secured through strong authentication and authorization, comprehensively auditable, and subject to sanctions for policy violations.

[***] CfH 2004b

- Secure and confidential treatment of patient information shall be a fundamental property of all technological and process artifacts pertaining to RLS. This tenet shall be built into the RLS architecture.
- Given the varying privacy regulations existing across states in the US, RLS should support the most stringent protection of health information, including the requirement of explicit patient consent for disclosure of data.
- In some cases varying levels of protection are required for different categories of clinical information pertaining to conditions such as mental illness and AIDS.

¶ **Decentralized and federated architectures**: Respecting the mandate for a de-centralized, federated architecture of healthcare information networks, RLS shall employ federated information architecture ensuring that each node in a RLS connected network retain 'informational sovereignty"[†††].

- A central data repository of aggregated patient healthcare data creates a large target and poses an unacceptable privacy risk in the current political and network environment.
- A 'National Health Identifier' for each person is impractical in the decentralized world of healthcare in the USA. Instead, RLS would support linking of health records that remain distributed and managed at their points of origin.
- Data distribution should be biased to local control of clinical records and access to them. Personal health information should continue to reside where they do now, primarily with hospitals and healthcare providers. Decisions about disclosure of such information should be made at the source of the data, with patient consent if so required.
- Users shall be authenticated and authorized to access patient information at the "edges" as well, obviating the need for centralized identity management of all authorized users.

¶ **Open Standards**: All solutions / components shall be based on open standards and not be dependent on any proprietary technologies. While standards in themselves do not guarantee interoperability, emphasizing standards across the network communication stack helps mitigate most of the common problems that have impeded information sharing in the past.

- HL7 (version 2.x or 3.0) and NCPDP SCRIPT, the de facto industry messaging standards for general healthcare and prescription data respectively, should be used where applicable. The HL7 Reference Information Model offers a ready set of data standards that provide the semantic interoperability underpinning for HL7 messaging standards.
- XML 1.0 is the data message notation standard for inter-application data communication and shall be the default message serialization format.
- New or private network services shall not be required: the solution should be based on secure data transport over the public Internet.

[†††] CfH2004a

Secure Sockets Layer (SSL) protocol, the most widely adopted security protocol standard for the Internet, shall be used.

- 'Web services' are the industry standard for platform-neutral, distributed application interoperation over the Internet. Web services should be used to effect data sharing across the health information network.
- The Web Services-Interoperability Organization (WS-I) provides profiles to assure that web services built on disparate platforms have higher assurance of interoperability.

¶ **Vendor and platform neutral**: The RLS solution needs to be vendor-neutral to ensure wide-spread adoption of the architectural standards. As an extension of the open-standards principle, this principle stipulates that no dependence on specific vendor technology be introduced.

- Leverage commercial-off-the-shelf (COTS) master patient index solutions for the prototype, but develop the architecture in a manner that future adherents to the Common Framework can make their own build vs. buy decisions.
- Assume that comprehensive integrated application suites are potentially cost-prohibitive for wide-spread deployment across a range of healthcare systems from small family practices to large hospital networks.

¶ **Best Practices**: RLS service should be designed for agility and extensibility to meet varying regional clinical data exchange implementation requirements.

- Service-oriented architecture enables applications to be more flexible, and interface well with external facing web-services. Following the service-oriented approach, the RLS application should comprise loosely-coupled coarse-grained components that can be readily reused. However, the internal architecture of the RLS application is not itself relevant to the standards based external messaging that is the primary requirement.
- RLS should support existing regional and community health information networks as well as prescribe best practices and patterns for new networks to adopt.
- Variability across regional networks should be mediated through shared specifications based on the open standards as prescribed above.

¶ **Promote Widespread Adoption**: The following system constraints should be taken into account to enable rapid deployment of a solution that builds on the RLS prototype:

- Widespread distribution of the RLS solution demands a light-weight inexpensive solution for all new components at the edges.
- Participants have diverse vendor relationships and must not be bound / committed to any *one* vendor to benefit from RLS-based connectivity. RLS specifications and standards should be open to implementation by healthcare information technology vendors.
- Standards based, loosely coupled, and flexible design should be used to avoid 'rip and replace' implementation across the current healthcare landscape. The architecture should support an

incremental migration from current technology to future standards based interoperation.

- Participants in health information networks have significant investments in EHR and personal health record (PHR) applications. RLS should support connectivity between them, with incremental migration paths that call for moderate incremental investment.

¶ **Flexible Implementation Models**: The RLS architecture should enable top-down and bottom-up implementation strategies, favoring local network infrastructure development to promote widespread usage and national interoperability models to promote inter-regional information sharing. From a technological standpoint, the RLS architecture should support three implementation models:

- Gateway: Physically deployable service that may be integrated into RLS subscriber enterprise IT environment, and allowing secure access to RLS.
- Application Programming Interface specifications: Allow solution providers (package vendors and custom development shops) to implement RLS components independently on platforms of their choice without detracting from interoperability.
- Hosted: All RLS services are hosted for subscribers unable or uninterested in owning and operating the RLS access infrastructure. Such a solution would be an attractive option for smaller physician practices and institutions that prefer to outsource their information technology services.

3 Use-Case View

The RLS system architecture is primarily driven by the functional requirements of the health information network. Functional requirements are captured in use case models or requirements definition documents, as well as vision and mission statements describing the longer term strategy. In addition, architectures have technology drivers and need to be cognizant of constraints of the information processing environment in which the system operates and the technology platforms used to implement the architecture.

The Use Case view represents the end users view of the system, and provides insight into the business goals of the system. Use cases represent the interactions that take place between the system and its users. Use case diagrams provide a schematic view of the end-user requirements that the system expects to satisfy. Use cases do not capture all the functionality of a system, only the users' activities with respect to the system. Also note that use case specification diagrams need to be supplemented by textual descriptions that provide details of the requirements, which are *not* provided in this document.

3.1 RLS Functions

RLS is intended to serve as a key infrastructure element in a regional health information network. RLS primarily maintains an index of pointers to the network location of patient information, but not the personal health information itself. The index of pointers is akin to a 'master patient index' as deployed in integrated healthcare delivery networks with multiple independent clinical systems maintaining patient records.

RLS serves as a coordinating service that obviates the need for a national health identifier, by linking diverse patient records across distributed clinical data sources through probabilistic demographic matching techniques. Such a service can facilitate a federation of diverse clinical data sources to enable a consolidated view of a patient's electronic health care records. The clinical data nomenclature includes a range of information from medical records in provider systems, dispensed drug information at pharmacy systems, to administrative and financial information in payer systems.

Various institutional models have been discussed for ownership and operation of an RLS. In the current environment, the logical organizational framework for an RLS would seem to be a regional health information organization (RHIO). RHIOs are considered key elements of the strategy for constructing an interconnected and

interoperable network of networks that forms the national health information network (NHIN).

Massachusetts SHARE (Simplifying Healthcare Among Regional Entities), a regional collaborative initiative operated by the Massachusetts Health Data Consortium, may play the role of a RHIO. One of MA-SHARE's goals is to create a sustainable community utility for clinical connectivity and data exchange. The future extension of RLS to serve as such a utility is a consideration in its architecture.

While RLS primarily supports individual caregivers by facilitating access to aggregated patient information, it may facilitate information sharing across other authorized stakeholders as well, including the patient. A longer term 'concept of operations' view of RLS is shown in Figure 2.

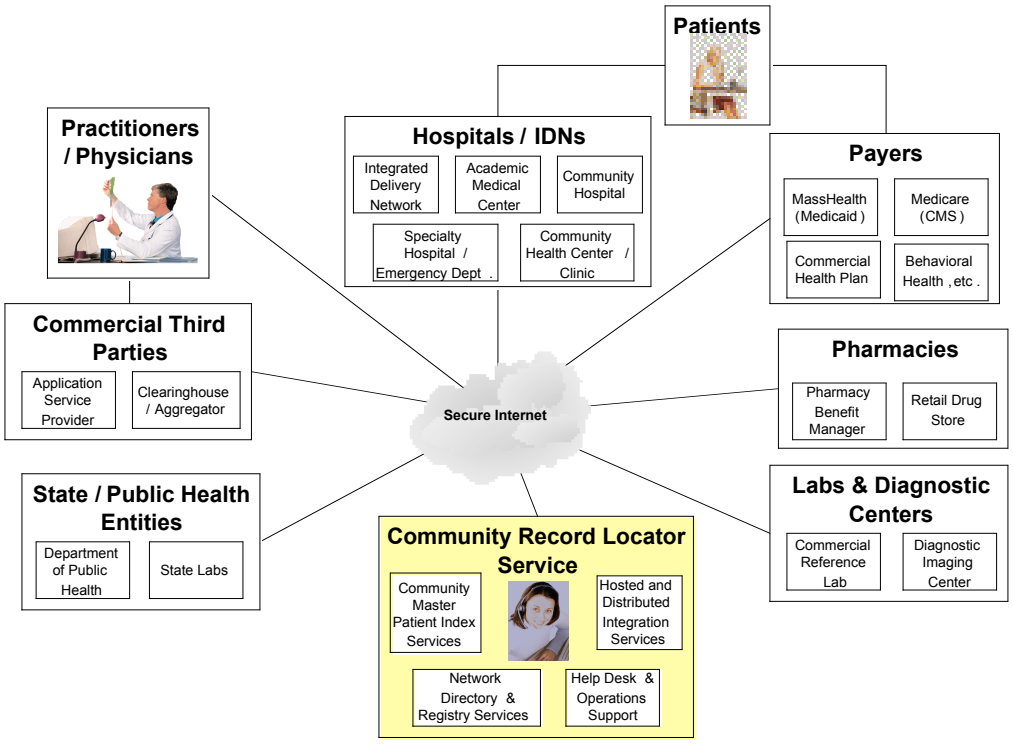


Figure 2: RLS Long term Concept of Operations

3.2 Use Cases

The architecturally significant use cases of RLS are shown in Figure 3. The primary end user function of RLS is to provide healthcare practitioners with pointers to clinical data stored at distributed network nodes. To establish context it useful to understand the complete usage scenario from a healthcare practitioner's perspective, which would include subsequent retrieval of patient records. Patient records retrieval is *not* an RLS function, and is

shown as a separate 'clinical data exchange' service in the use case diagram.

Actors are entities (both human and system) that interact directly with the RLS. RLS actors include:

1. Healthcare practitioners: Individual care providers who have been assigned rights to access a patient's clinical record. This category includes providers, payers, diagnostic services etc., as well as the patient.
2. Clinical Data Source: The information systems in use by a healthcare provider or payer to maintain patient related information. Data sources encompass systems at physician offices, hospitals, laboratories, imaging centers, pharmacies and other healthcare service entities.
3. RLS Administrator: Persons (or entities) authorized to administer the Record Locator Service and the community

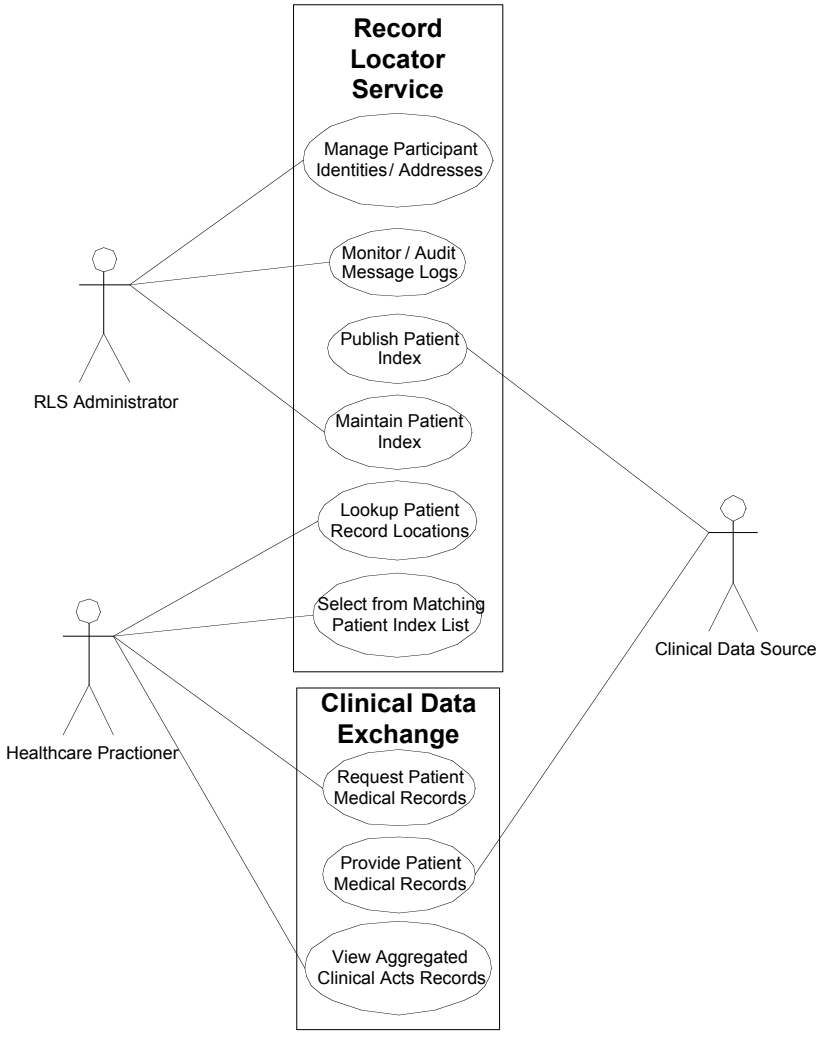


Figure 3: RLS Prototype Use Case Diagram

patient index as may be required. Note: this role is expected to be progressively automated as the RLS implementation matures.

Note that the core RLS function is only to locate patient records. This essentially implies providing indexing services for distributed clinical data sources, which publish patient registry events into the RLS. This separation of services is a key aspect of the CfH strategy which seeks to decouple the standards and policies relating to separate functions of the network[‡‡‡].

The following use cases are architecturally significant to the RLS, in that they exercise critical aspects of the system architecture:

Table 1 List of Architecturally Significant Use Cases

Name	***Description***
Lookup patients	On entry of search criteria by authenticated and authorized users, system shall retrieve a list of patients matching the search criteria entered. System shall return the list of patient records with 'locations' (or web addresses) where these records may be accessed
Publish patient index	On entry of new patient records in the clinical data source (e.g. after a registration or ADT event), or upon changes to existing patient records, the clinical data source node shall transmit a set of demographic attributes and a pointer to the patient record (typically in the form of a Medical Record Number) to the patient index maintained centrally at the RLS (the community Master Patient Index, or CMPI) The RLS acknowledges receipt of the changed record information.
Authenticate authorized users	RLS users are authenticated as authorized network users by the clinical system to which the user is affiliated.
Communicate securely	RLS and the clinical system communicate securely over the internet. Senders and receivers of messages are mutually authenticated before exchange of messages; message confidentiality and integrity are assured; and message non-repudiation is enabled for both sender and receiver
Log messages	All messages are logged with name of user / organization initiating the operation. Logs can be audited for information on all access to RLS patient index

‡‡‡ CfH2005b

3.3 Use-Case Realizations

The patient lookup and patient publish use cases are both realized through messages sent securely from nodes in the health information network to the RLS requesting lookup and publish services respectively. The 'lookup patient' and 'publish patient index' use cases are realized in the manner shown in the activity diagram in Figure 4.

The swimlanes in the activity diagrams correspond to the actors shown in the use case diagram. Healthcare practitioners and clinical data sources are roles played by entities such as hospitals, physician practices, diagnostic services, payers, public health agencies etc. Thus, a provider system at a network node could play the role of a clinical data source, and also be the channel through which healthcare practitioners access the RLS.

While the core functionality of RLS is to support publishing into, and searching the CMPI, the RLS-based network would require information processing in each of the nodes interacting with the patient index to support secure, reliable and standards based communication between them. This communication function is the core of the common framework that enables the various network nodes to exchange information with the RLS and with each other. The communication functionality at these nodes share many common processing capabilities which may be encapsulated into a common technology artifact as shown in Section 4.

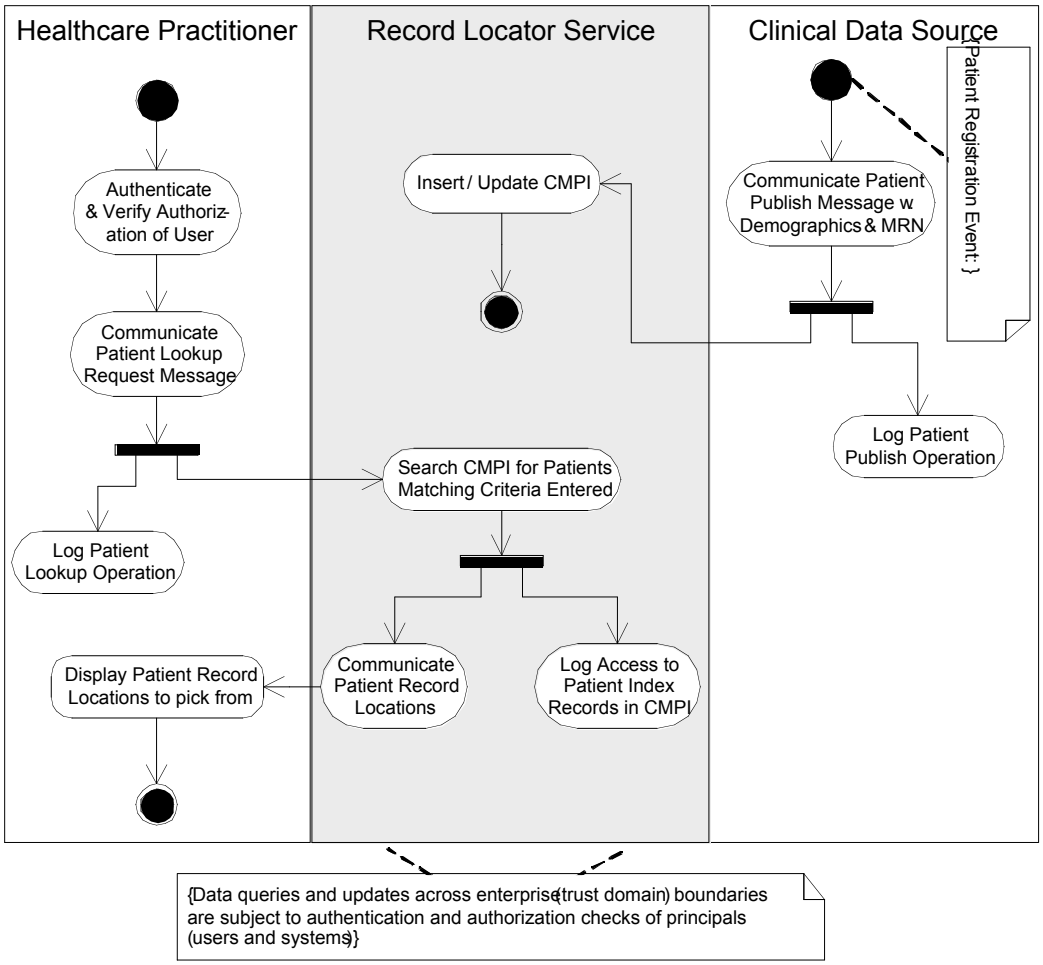


Figure 4 Lookup Patient and Publish Patient Index Activity Diagrams

Other functional aspects that influence the RLS architecture include:

3.4 Security, Patient Privacy and Consent Management

Protection of patient privacy is a legal and regulatory requirement that is realized through system security as well as policies implemented by the RLS and the various participants in the healthcare information network. Patient and healthcare practitioner trust in the security and privacy protection features of the RLS-based network is a critical pre-requisite to its success.

Privacy protection is based on restricting network access to the data to only authorized personnel, monitoring all access to patient data to audit the operational use of the network, and implementing architectural safeguards to manage the risk of accidental or malicious spillage of data in the course of network operation.

3.4.1 Identity management

The RLS is intended for use by large networks of healthcare enterprises, each of which may have large user end user populations. RLS must not be required to manage the network identities of all individual users. End users (healthcare practitioners) are authenticated by the enterprise to which they are affiliated and referred to RLS as authenticated principals. Identity management is a local function. Patients who use hospital / IDN patient web sites are referred to RLS as authenticated principals by the clinical systems at the hospital / IDN. RLS and all participating entities have the ability to mutually authenticate each other before exchanging data.

3.4.2 Confidentiality, Authentication, Integrity & Non-repudiation

The RLS network architecture uses the following high level security requirements in implementing messaging as well as application data management[§§§]:

- Confidentiality: Information shall only be disclosed to authorized users who need it for healthcare treatment, payment, or operations.
- Authentication: Receivers of requests for information shall be able to verify the identity of the requester. A network participant shall not be able to masquerade as anybody else.
- Integrity: Communication between network entities shall be protected against unauthorized alteration, and all alterations shall be logged. Receiver shall be able to verify that the message has not been altered.
- Non-repudiation: Transactions cannot be unilaterally revoked or altered by either party. A sender cannot falsely deny sending a message and a receiver cannot falsely deny receipt of a message.

3.4.3 Patient Data Privacy

The health information network is architected on the presumption that data privacy is easier to protect locally, i.e. on the edges of the network where data are stored. Release of information from the clinical data source to healthcare practitioners is governed by policies established and maintained by the data source.

RLS, being a directory of patient record locations, is itself a source of protected health information. RLS shall publish and maintain a clear policy governing discovery of patient information in the directory. The specific rules governing sharing of RLS directory information pertaining to individual patients are created by the patient and provider at the time of the encounter and communicated to the RLS along with the message of that encounter event.

Healthcare information networks need to pay special heed to 'sensitive' data disclosure:

[§§§] CfH2005b

- There are categories of medical information that need additional safeguards which RLS should be cognizant of. These include mental health, AIDS, and substance abuse related care data.
- While RLS does not itself retain patient care data, the availability of patient records at a mental health facility is itself disclosing. RLS access control policies should be informed by such considerations.
- The patient consent process should be capable of allowing the patient to set varying restrictions on the different categories of sensitive data.
- A 'break-the-glass' function should allow authorized providers to override patient privacy constraints in emergency situations that require access to sensitive data.

3.4.4 Consent Management

Some states require that explicit patient consent be obtained to share their medical records across the network. There are varying degrees of restrictions of privacy / consent requirement and RLS must be capable of handling this variability. The following policies have been proposed for the RLS-based health information network to manage the consent process:

- Require that patients be fully informed in writing of a provider's or health plan's participation in the RLS before their information is exchanged through the network;
- Mandate that the written notice given to patients contain certain disclosures about how information is used and exchanged through the RLS
- Permit providers and health plans to include the disclosures about the RLS in their HIPAA privacy notices
- Give each patient the right to decline to have their information exchanged through the RLS ("opt-out"); and
- Prohibit providers and health plans from withholding treatment or benefits to patients who have opted out.

3.5 Patients Records Linking and Matching

The key function of the CMPI is to link patient records across different institutions, each of which maintains patient data independently and, often, inconsistently. Various algorithms are available for matching person records based on limited sets of demographic attributes, for which software implementations exist. The algorithms should match the patient attributes used as search criteria with those in the CMPI records using NYSIIS matching, digit transposition checks, etc. RLS should be capable of using integrating with a variety of patient matching software.

Two models exist for implementing the linking / matching process.

- Patient records may be linked when they are loaded based on running a matching algorithm with the rest of the patient records in the database. An online query would then be matched using the same algorithm to the pre-linked records. This process is typically supported with some level of human disambiguation of data. For

example a data administrator may examine records that match marginally and manually ascertain a positive or negative match. Note: this function could potentially grow into a large data maintenance organization.

- Matching is done during a RLS patient lookup query, but no explicit linking of records is done in the CMPI database. Very fast database matching speeds have been achieved using probabilistic algorithms that also provide high assurance of no false positives. A disadvantage is that this completely automated process could potentially result in higher false negatives. Patient privacy considerations (no false positives) imply high thresholds for probabilistic matching, which would miss patient records that match marginally.

Human disambiguation of patient matching results is considered undesirable. While an interactive narrowing down of patient matches may be appropriate within a hospital setting where the degree of trust is high, this is considered unacceptable when RLS is serving as a CMPI to multiple institutions. Therefore, probabilistic matching algorithms should have their positive match threshold set high enough to reduce the percentage likelihood of false positives to negligible levels.

- RLS patient lookup service shall not support wild-card matching. This could open the door to database fishing which would be an egregious violation of patient privacy principles.
- RLS should support the retrieval of record locations using a local patient ID.
  - There could be situations where a patient is well-identified locally (e.g. has a long-established MRN). It should be possible to get the linked records from RLS rather than do a demographics attributes based search
  - This may technically be considered a subset of the demographics based search. RLS maintains both patient source institution and institution local MRN.

4 Logical View

The architecturally significant subsystems / components that make up the RLS are identified and their interactions that provide the required services are described in this section. A top-down approach is taken starting with the high level functional component view, and decomposing this into more granular components that can be translated into system components and services.

4.1 Conceptual RLS-Services View

Based on the activity diagram that defines the RLS patient lookup process it is clear that functionality is required at three processing nodes. These are the healthcare practitioners, the clinical data sources, and a patient index.

A conceptual view of the interaction between RLS and its 'subscribers' may be viewed as shown in Figure 5[****].

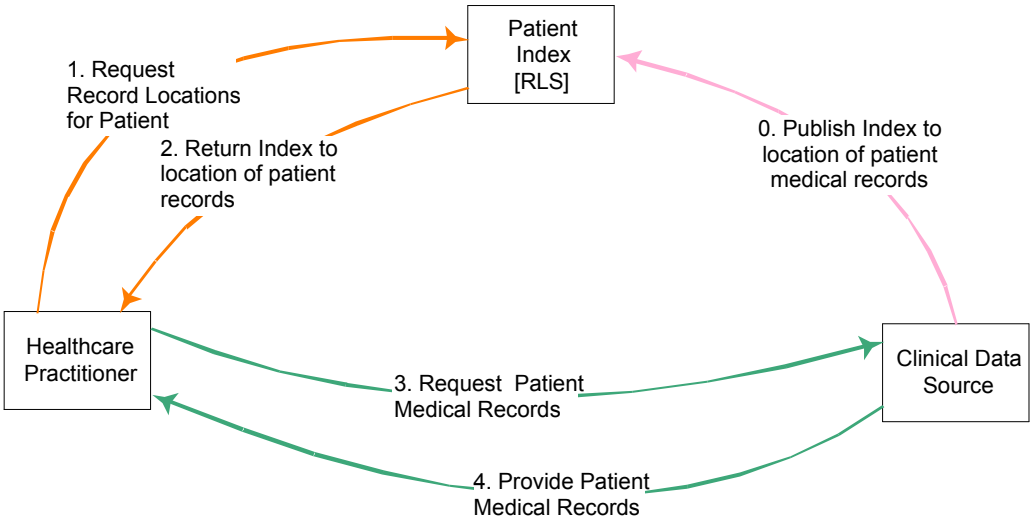
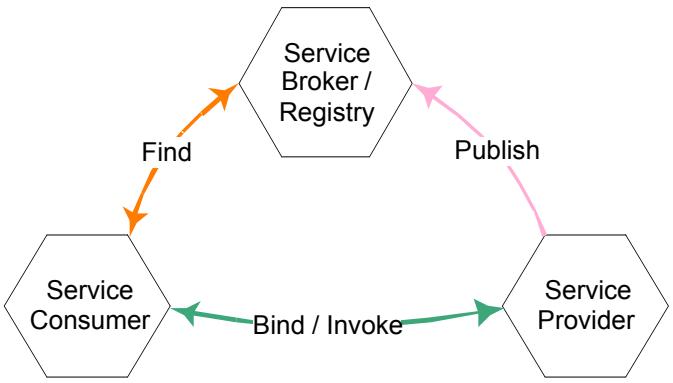


Figure 5 RLS Conceptual Architecture of Operation

By maintaining the patient index pointers to patient record locations in multiple clinical data sources systems, RLS serves as a directory of patient records in clinical systems. Besides the patient index, the RLS also maintains the registry of all members of the network, and their network addresses.

This model is seen to very similar to the classic service-oriented invocation paradigm where the service consumer, provider, and broker interact in the 'publish / bind / find' pattern, as shown in Figure 6. This similarity suggests that RLS should play the role of a patient record registry and that clinical data sources should be

[****] [CfH 2005a]



Fiaure 6 Service Oriented Interoperation

exposed as services that are accessible via open standards based messaging interfaces.

Service oriented architecture (SOA) is an architectural style that promotes system agility and extensibility through loosely-coupled software components interoperating through generic messaging interfaces. Interfaces may be separated from implementation by expressing application semantics through XML messaging interfaces. Extensibility should allow new versions of the service to be published and consumed without breaking the existing service, which is facilitated through XML Schema versioning.

Web services are implementations of SOA across enterprise boundaries where interfaces use Internet transport protocols (HTTP, SMTP, and FTP). Web service communications commonly use SOAP protocols for message packaging and WSDL for service description. Service brokers and registries are also accessed through SOAP messages. Thus, in keeping with the architectural principles set out in Section 2, RLS should be implemented as an Internet accessible service using standard protocols such as SOAP and WSDL.

In addition, RLS should use healthcare domain data standards that support semantic interoperability with the various clinical systems. This would expose the resources managed by RLS (i.e. patient index) to consumers in the network through a standard representation based an industry standard information model, such as the HL7 RIM. HL7 v3 messages are derived from the RIM and, therefore, have intrinsically better semantic interoperability characteristics than earlier versions.

There are practical difficulties in implementing the above open-standards messaging interface based interaction pattern in a healthcare information exchange setting:

¶ While a large number of clinical systems support HL7 2.x messaging and interface with other systems within enterprise networks, message implementations are not consistent and inter-enterprise data sharing

is difficult. Use of HL7 RIM is not widespread in the industry and HL7 v3 implementations are extremely scant.

- Most clinical data sources do not conform to inter-enterprise interoperability messaging standards, and very few are 'Web service enabled' to consume or provide Web services.
- Canonical message formats are essential to support practical peer-to-peer information sharing. Otherwise, in a network with n nodes sharing data, it is likely that the number of distinct translations for *each* message exchange (request/response) is of the order of: $n \times (n-1)$.

RLS provides the master patient index service to locate patient records at distributed clinical data sources, components to interface with each of the clinical data sources, and canonical data formats for the patient index publish and lookup messages.

4.2 RLS Application Services

The common framework does not place any requirements of the RLS internal application architecture. As long as the external RLS interfaces conform to the proposed messaging standards, interoperability does not depend on the implementation details of the service. Nevertheless, the following discussion provides guidelines based on the experiences with RLS prototype development that are expected to be useful for other RLS implementation projects. As the discussion below shows, the use of SOA principles enables flexible implementation models, and network topologies while conforming to essential interoperation standards, which is a primary requirement of the common framework.

As discussed in Section 4.1, services are application components that expose their functionality through standard interfaces. In addition, the service model is fractal in that services may be created by combining other services to expose new, aggregated capabilities. Such an aggregation is also called an 'orchestration' or 'composition' of services. The 'composability' of services is important to the agility of SOA applications, and to understanding the RLS application architecture.

Following the service-oriented approach, the RLS application is logically structured as an aggregation of coarse-grained loosely-coupled services. In systems architectures it is useful to classify services as business services or infrastructure services. Infrastructure services are reused across multiple business services, enabling cost effective systems development and operations. The RLS is composed of multiple services, both business and infrastructure. The core business service provided by RLS is:

- Patient index service
  - Responds to patient lookup queries with a list of patient record locations

- Accepts patient index (record location) updates from clinical data sources.

RLS business services are supported by common infrastructure services such as:

- Security: Services that handle user identity management, authentication and authorization, protection of patient privacy, and consent management.
- Systems management: Covering automated administration, installation, configuration, and operational monitoring, control and optimization of systems.
  - Logging: To meet auditing and system maintenance requirements
- Data services: Provide persistent storage and management of data, as well as common data access mechanisms for application components.
- Message transport: Enables reliable, synchronous, message based communication between system components.
- Message transformation: Overcomes the real-world problem of disparate data format standards supported by different systems.
- Web-services interface: Leverages the numerous WS-* standards to expose RLS services through XML based messaging API accessible over HTTP transport networks.

Web service interface services may be categorized as belonging to certain standard architectural patterns, called 'service gateway' and 'service interface'[††††]. The service gateway is an agent, encapsulating the details of communicating with remote web services, and enabling legacy applications to consume web services. The 'service interface' is a façade, encapsulating the legacy application by overlaying a web services wrapper, and enabling remote systems to communicate to it.

Technically, RLS can be implemented as a composition of the above services to provide services to remote applications over the globally available Internet. But in recognition of the practical constraints of legacy clinical systems, the architecture also provides guidance on connecting clinical systems to RLS. This may be accomplished through a web service interface as listed above and:

- Adaptors: Facilitate connections to clinical systems and databases through database interfaces or custom API

The message transformation, web services interface and adaptor components are the key infrastructural service for the interoperation of disparate clinical systems. Essentially, these infrastructure services expose the RLS patient index as well as clinical systems as web services, and enable them to consume web services. Thus connectivity in the network is established using platform and payload agnostic, open standards.

[††††] Service Patterns, Version 1.0.0, Microsoft Patterns and Practices, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpatterns/html/EspServicesPatterns.asp?frame=true, 2005-03-31

4.3 Gateway Services

Following the principle of deploying applications as aggregations of loosely coupled services, the infrastructure services listed above could be bundled to form a composite 'Clinical Data Exchange (CDX) Gateway' for the clinical systems to interoperate with each other. By designing the CDX Gateway to support general purpose, secure, and reliable messaging between clinical systems, it serves as the standards-based on-ramp to a health information network.

The RLS may be realized as the orchestration of the following two service compositions, which may be considered as the business service and infrastructure service respectively:

- Patient Index: Central service that maintains the community patient index, and a registry of clinical systems with routing information to direct service requests and responses to appropriate end-points.
- CDX Gateway: Distributed service that interfaces to each network node, converts from the legacy data messaging format to the standard message (if needed), and communicates to other gateways in the network through Web services.

The CDX Gateway needs to be a small footprint, low-cost service that can be deployed in participant institutions with minimal customization to interface with disparate clinical systems on one side, and to other Gateways via the public Internet on the other. A CDX Gateway is also deployed at the RLS to support messaging with the other network nodes, exposing the Patient Index to external systems through this common utility component.

Such a gateway serves as a general purpose utility to support medical records retrieval as well as RLS communication services. Users acquire patient record locations (pointers) from the RLS and access data from multiple clinical data sources as shown in Figure 7. Note the parallels with the conceptual architectural vision in Figure 5.

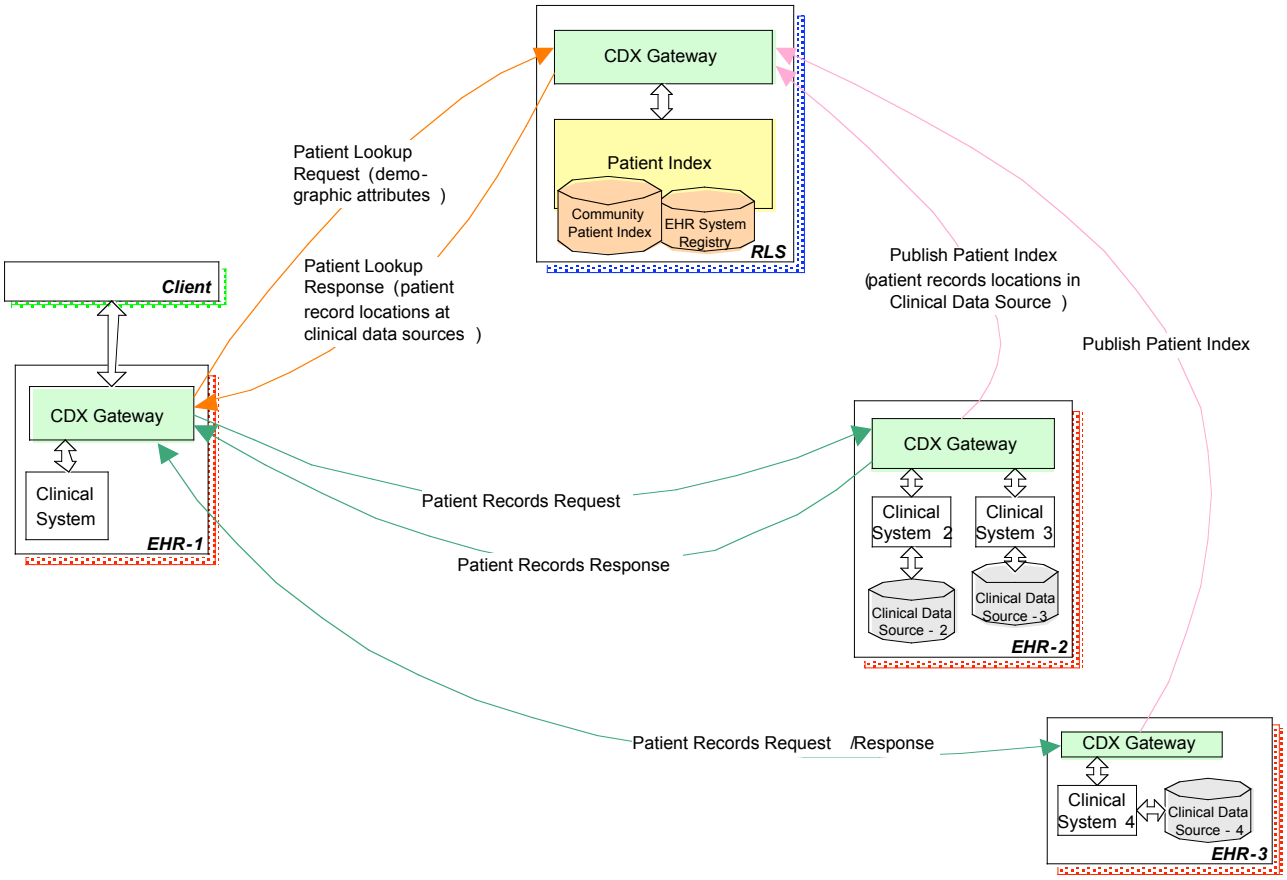


Figure 7: RLS Distribution of Components

As may be seen from Figure 7, communication between all network nodes is through the CDX Gateway at that node. In addition, presentation and business services would be required for the user interface functionality of the RLS (patient search criteria entry, and selection of patient record locations to query). Online users would log into the CDX Gateway co-located with the clinical system to which they are affiliated and access RLS services.

The collection of clinical systems that hold patient medical information at each network node is called the Electronic Health Record (EHR) in the figure, and in the following discussions. In the figure an online interaction is shown at the CDX Gateway at EHR-1, whence the patient lookup request is made to the RLS. The other nodes EHR-2 and EHR-3 serve as clinical data sources. Data retrieval from backend systems at the clinical data sources is done through the adaptor services in the CDX Gateway. The data sources publish patient index updates to the RLS also via the CDX Gateway services at those nodes. Although not shown in the figure, online users at EHR-2 and EHR-3 can also access patient lookup services at RLS (and other clinical data exchange services).

RLS receives service requests through a CDX Gateway at its location. RLS needs an administrative user interface to manage the application, which would be the responsibility of the presentation and business services in the gateway at the RLS.

The gateway based architecture provides significant flexibility and scalability. More clinical data sources could be added by deploying a CDX Gateway at that location and customizing adaptors to connect with the clinical system there. The Gateway transforms local message formats into standard ones and manages their secure, reliable communication to other Gateways.

In general, the expectation is that CDX Gateways that are approximate clones of each other would simplify deployment, configuration and administration of this distributed service. The CDX Gateway, thus, serves as a general utility service that may be plugged in at different EHR locations. In addition the CDX Gateway serves as an infrastructure component which realizes the common framework standards in a packaged form.

CDX Gateway application architecture is based on a multi-tiered pattern of presentation, business, and data and integration tiers. The integration tier is the service that interfaces with clinical systems / data sources at the backend, and communicates with remote Gateways through orchestrated web services at the other. Gateways also include capability to serve as messaging intermediaries. This is a side-effect of implementation of WS-* standards which include WS-Security and WS-Addressing in the CDX Gateway that enables secure, reliable SOAP intermediary functionality.

A more detailed view of the CDX Gateway and its interaction with other gateways is shown in Figure 8. The distribution of functionality between a pair of communicating Gateways is flexible, dependent on the capabilities available at the clinical systems at each network node. The architecture shown in Figure 8 depicts the two EHR nodes playing distinct roles: healthcare practitioner and clinical data source. There are no backend clinical systems at the healthcare practitioner node and the presentation and business services are not deployed at the clinical data source. Within the gateway, the message handling, clinical system adaptors and web service interfaces are shown encapsulated in an Integration Broker service.

It is expected that each CDX Gateway deployment initially requires significant custom localization of the Integration Broker service to cater to EHRs that do not conform to prescribed standards. Over time clinical systems are expected to develop standard Web service interfaces. This would reduce the processing requirements of CDX Gateway, and would enable lighter weight Gateways to be directly pluggable into the data sources.

CDX Gateways communicate with each other using Web services protocol (WSDL and SOAP) over the standard HTTP/SSL transport protocol. Gateways could extend in future to adopt alternate transport protocol such as Secure FTP for batch file transfer, and SMTP using industry standard S/MIME encryption for email.

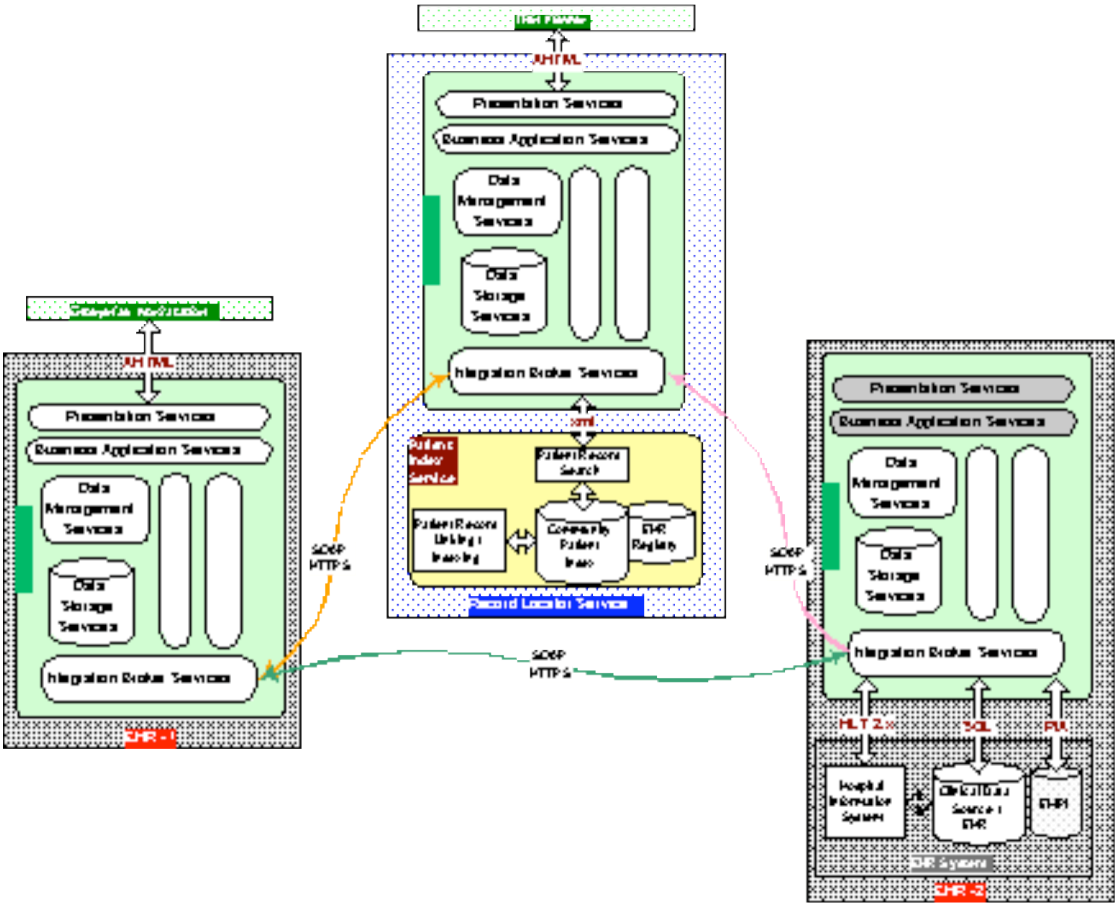


Figure 8: Gateway based interaction in a health information network

4.4 RLS-Based Networks

The RLS-based network strategy may be summarized as: Web-service enabling clinical systems. Nodes communicate with each other through HL7 (or other domain-specific standard format) messages wrapped in SOAP envelopes over HTTPS transport. Such a peer-to-peer clinical information network is shown in Figure 9.

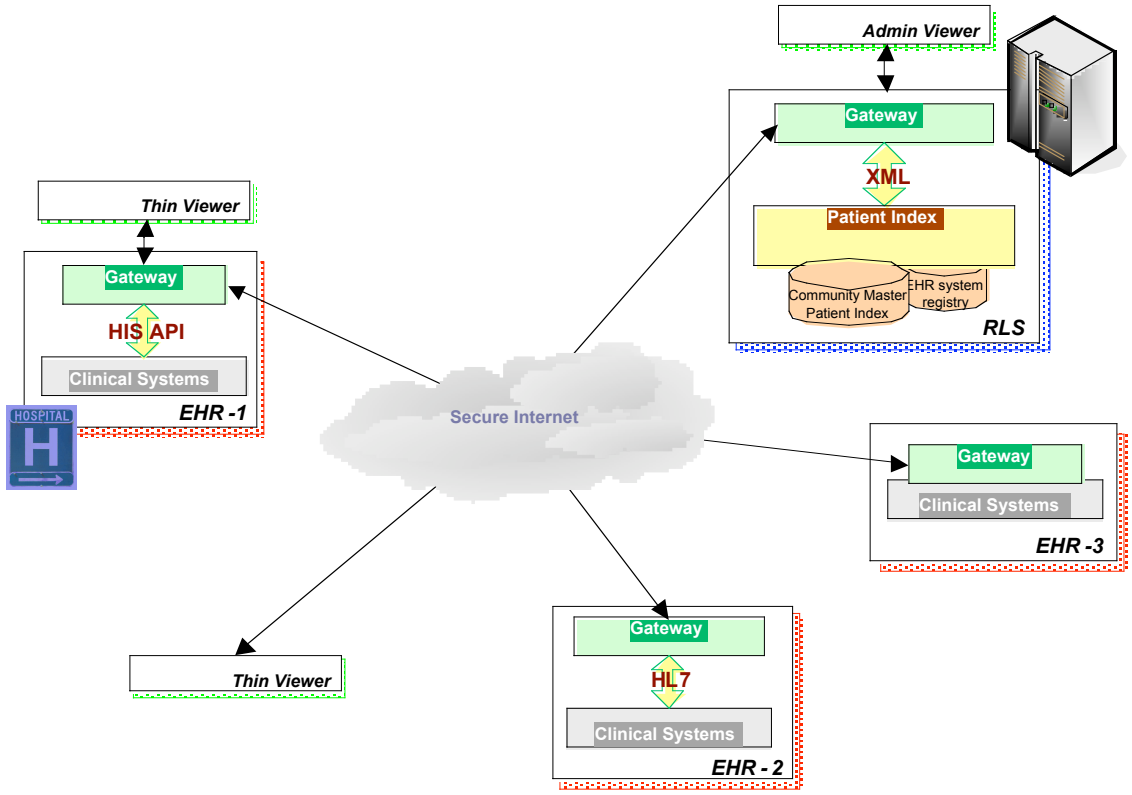


Figure 9 Network of clinical systems communicating peer-to-peer

The diagram shows diverse ways of deploying CDX Gateway to connect health information network participants to RLS. While clinical systems routinely send patient index updates to the RLS, users connect to RLS, via their local clinical system, to query patient record locations (patient lookup). Patient lookup can also be done in a non-interactive manner through request / response messages exchanged between gateways at EHR locations and the RLS. In addition a user can query RLS from anywhere on the Internet using only a thin viewer. This mode requires the CDX Gateway at the RLS location to serve up the user interface and business logic much as a gateway at an EHR location would. Since CDX Gateways are clones of each other, this can be achieved by configuring the services in the RLS gateway appropriately.

In addition, it should be noted that the gateway service could play the role of a data cache. The data storage layer in the CDX Gateway (see Figure 8) could potentially hold a data repository into which clinical data is replicated from the EHR and made available to network access. Indeed, it is thought that clinical enterprises would likely prefer to have database queries be directed at such a proxy cache rather than exposing operational systems to remote queries over a health information network.

4.5 Regional and National Network Support

Current thinking on national health information network envisages interconnected exchanges hosted by Regional Health Information Organizations (RHIOs). Such a network of networks is shown in Figure 10.

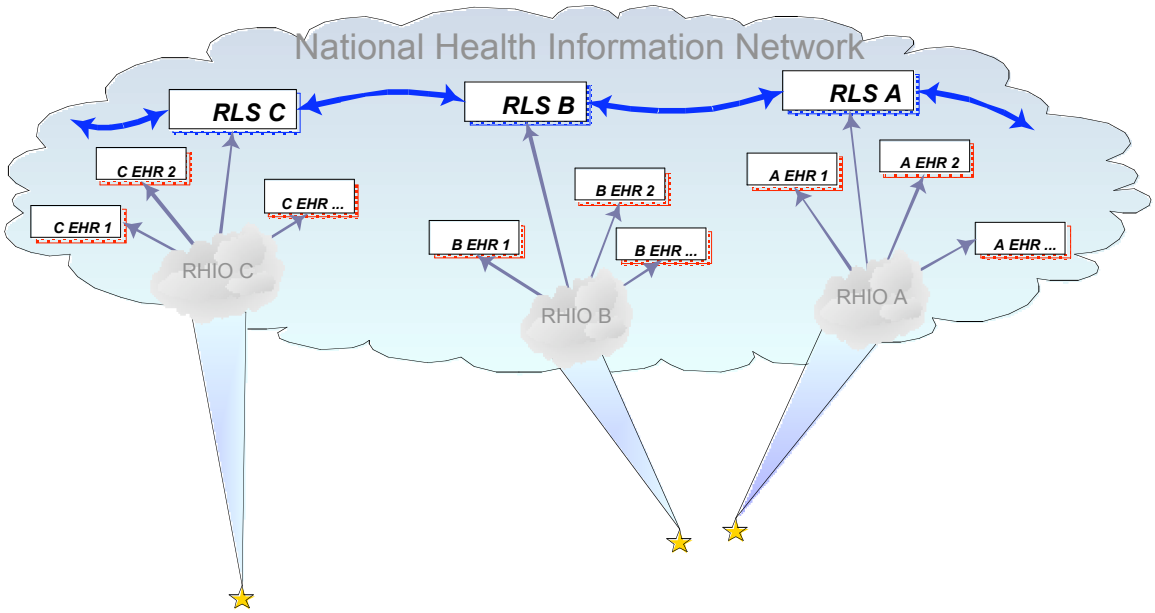


Figure 10 Network of RLS-based networks (potentially used by RHIO)

The RLS architecture itself presumes no regional dimension; the scoping function of the RLS is the community master patient index that maintains pointers to patient records in distributed EHRs in the community. However, RLS clearly is a candidate for the central coordinating service of a RHIO network. In regards to inter-RHIO communication, it is easy to see that the RLS could also play a key role in routing and coordination of services across RHIOs.

Some RHIOs may be based upon a centralized data repository as determined by regional policy considerations (as well as legacy architectural considerations); others are decentralized. Both models, as well as intermediate ones, are supported by the proposed NHIN architecture. Using a common framework based on open standards and common policies allows a RHIO to be agnostic about the architecture of another RHIO. The service oriented CDX Gateway architecture also enables a degree of flexibility in implementation styles and operational configuration as discussed further in the following section.

5 Process View

The RLS application is a collection of loosely-coupled interoperable services, and is itself a service that can be invoked by external consumers. Components that are distributed across network nodes communicate via XML messages using Web service standards, primarily SOAP and WSDL. Additional Web service standards may be used to support reliable messaging, error handling, and security.

The essential RLS functions to fulfill the patient lookup and publish patient index use cases are realized through distributed processing of the RLS services as described below.

5.1 Patient Lookup and Peer to Peer Medical Records Retrieval

The patient lookup process may be visualized as shown in the simplified communication diagram in Figure 11, where the flow of information and messages between the distributed processing components may be tracked through the sequence numbers provided.

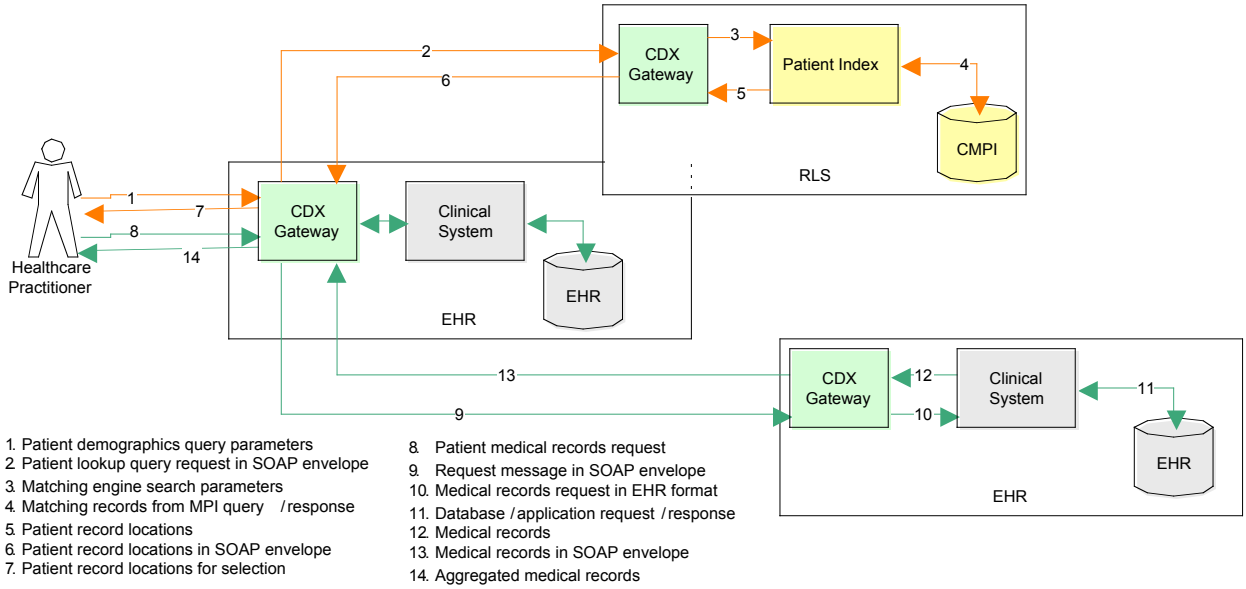


Figure 11: Patient Lookup with RLS and Medical Records Retrieval through CDX Gateways

The interaction diagram illustrates the use of gateways to manage all communications between network nodes. Gateways also encapsulate presentation, business, and data access services thereby providing a full application stack for flexible implementation of RLS services. For example, while the patient lookup is shown as an interactive process it could as well be transacted offline, in batch mode.

5.2 Patient Index Publish

The patient publish process is simpler in that it is a notification-type background interaction. The initiating message is triggered by a registration event at the clinical data source and essentially serves to update the patient index with details of the patient whose record has been updated, i.e. added, revised, cancelled, or merged with another patient record in the clinical data source. This interaction is shown in Figure 12.

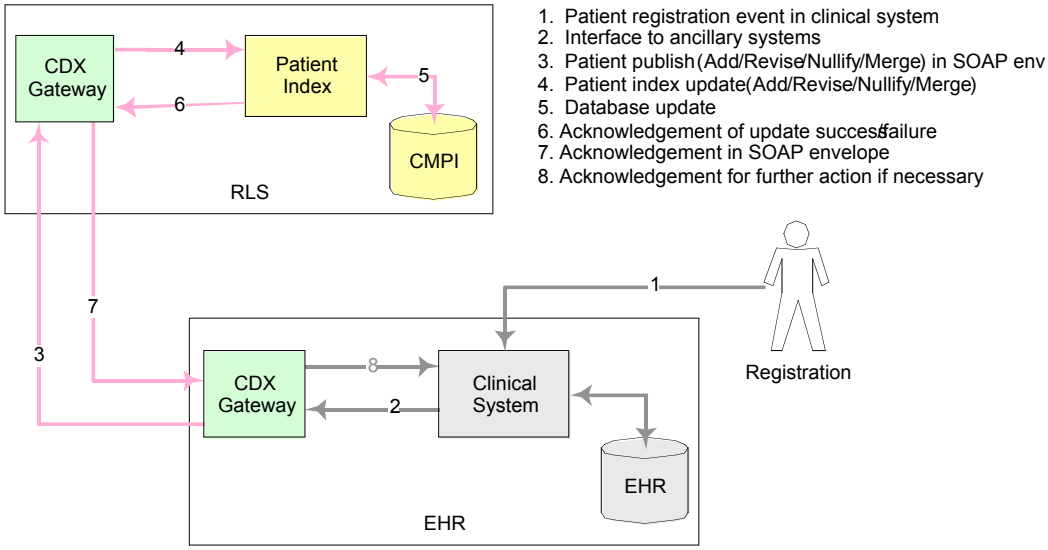


Figure 12 Patient Publish into RLS Patient Index

Typically, patient registrations are maintained in ADT systems in hospital environments. An ADT event message is broadcast to the other ancillary systems in the hospital such as laboratory, radiology, transcription, etc. The RLS patient publish message is generated by tapping off the same broadcast ADT message, requiring minimal changes to existing hospital systems. While this message is notification only, a basic exception handling capability needs to be built, where errors in posting the update to the RLS patient index are communicated to the data source. If the error does not require reentry of the ADT transaction, a mechanism to fix the data problem and resend the message to RLS needs to be built in the clinical system or interface engine.

5.3 Centrally Mediated Medical Records Retrieval

The scenario shown in Figure 11 represents a federated architecture, supporting peer-to-peer clinical data exchange. Alternate scenarios exist, such as a hub and spoke model where patient medical requests are mediated by a central gateway service. The collaboration diagram below demonstrates how a user could log in directly to a central CDX Gateway (in this case, co-located with RLS) and perform the same function.

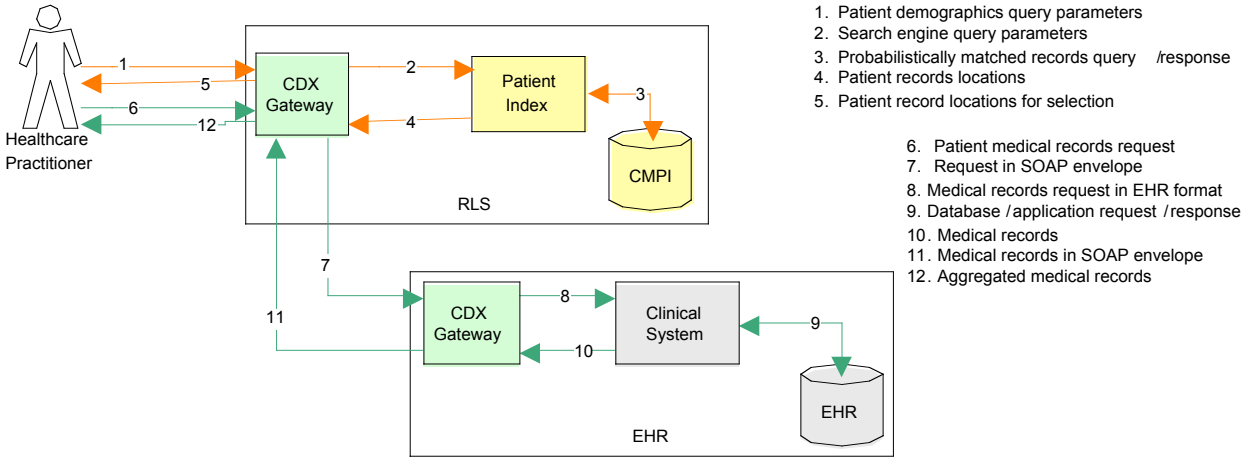


Figure 13 Centrally Mediated Patient Lookup and Record Retrieval (hosted gateway)

Note that the healthcare practitioner is shown as logging directly into the gateway at the RLS location and executing the patient lookup query directly on RLS. This is another variation of the use of the CDX Gateway, and demonstrates the implementation flexibility a standard gateway utility based network architecture allows.

5.4 Central Medical Records Aggregation

Yet another processing model would remove the need for a two step process altogether and have the patient lookup request be combined with a medical records request which the CDX Gateway at the RLS would orchestrate. Such a scenario is shown in Figure 14.

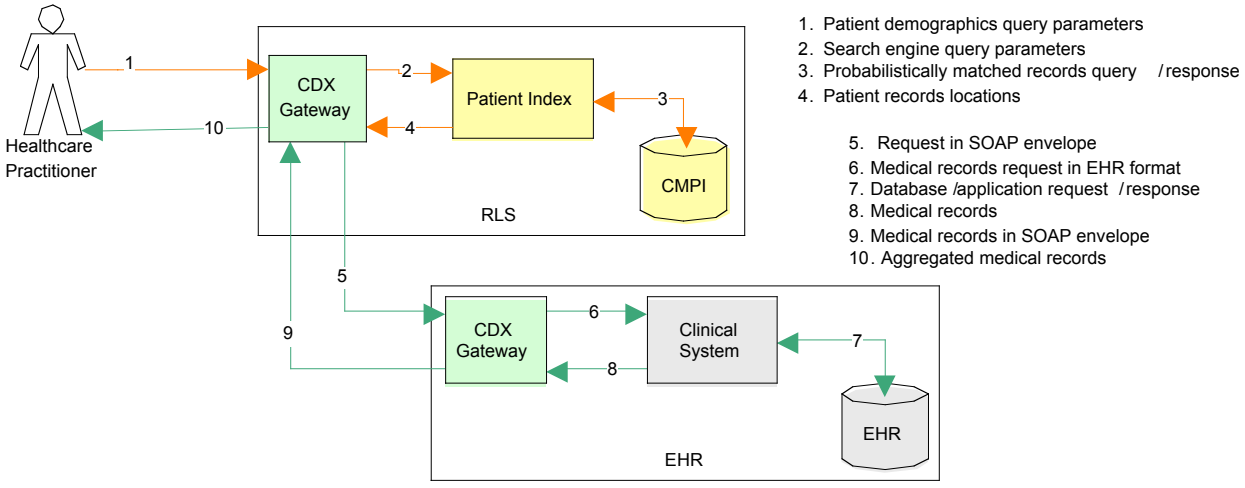


Figure 14 Patient Lookup and Records Retrieval -- In One Step

This interaction pattern is more appropriate for unattended RLS usage where a clinical system collects remote patient records and has them ready for review by a healthcare practitioner at later time. The orchestration process in the CDX Gateway is now more complex and should implement workflow processes of the healthcare practitioner, including selection of appropriate medical records to retrieve from various sources.

Other supplementary processes include the need for secure information sharing, and the establishment of contracts between the RLS and the various clinical data sources that would share patient medical records with each other. In both of these situations RLS, serves as a trusted intermediary, and reduces the need for the many-to-many relationships between the various clinical data sources. The secure messaging process is described below.

5.5 Security Processes

RLS needs a security framework to authenticate users, authorize access to specific services, and ensure confidentiality and integrity of messages. The implementation of security across a disparate, distributed computing network is optimally effected through a federated identity management architecture, where each user is authenticated by an assigned node that vouches for the user to other nodes. Federated security architecture is complex and considered beyond the scope of the current release of RLS. RLS uses a simpler model that can be extended to a federated architecture in later releases.

A distributed authentication/authorization architecture that is based on overlapping trust relationships is shown in Figure 15. Users' identity and credentials are maintained at the gateway that they log in to. Gateways are within the clinical enterprise domain, and may be integrated with the enterprise security infrastructure to support single sign-on for users. Gateways are in a trust relationship with other gateways and authenticate each other through server-side certificates. An authentication / authorization assertion is communicated in the SOAP message along with the user identifier string for audit purposes.

Secure Socket Layer (SSL) is a session layer protocol for sending encrypted information over HTTP. SSL provides an encrypted channel with confidentiality, integrity and one-way or two-way authentication. SSL is used to secure messages between gateways in the RLS-based network. Gateway authentication may be provided by server-side digital certificates.

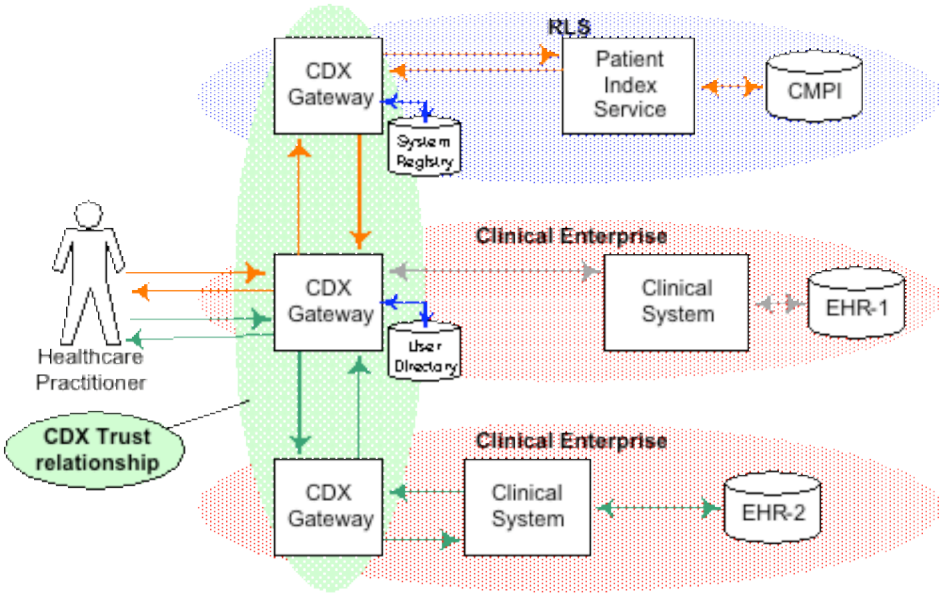
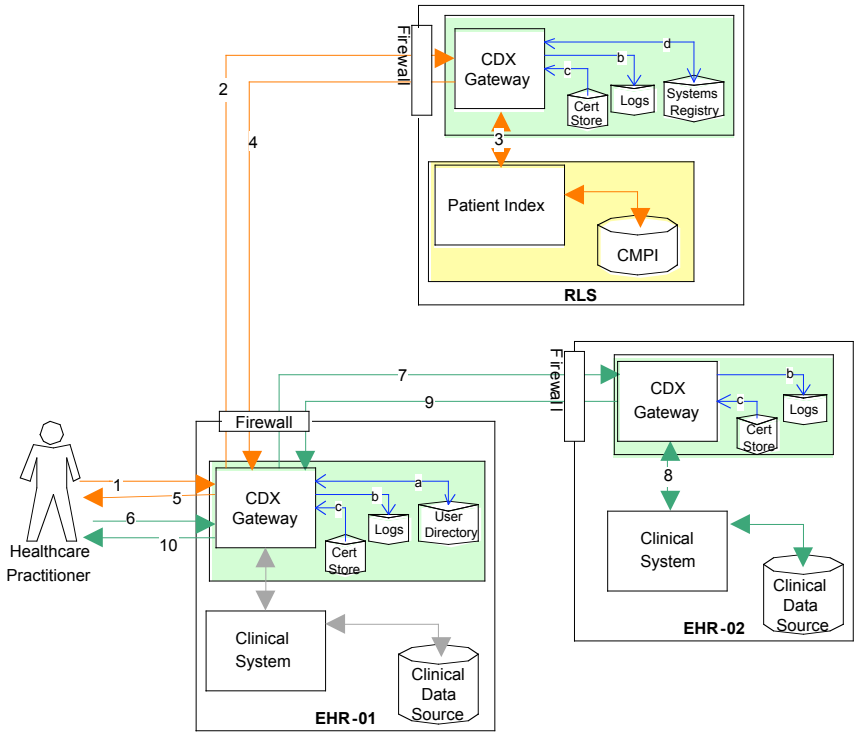


Figure 15 RLS authentication service

Over the longer term, CDX Gateways should leverage WS-Security, the emerging security standard for SOAP messaging, where different security tokens are embedded in SOAP Headers. The XML Signature and XML Encryption standard provides a platform neutral approach to message-level authentication, confidentiality, integrity, and non-repudiation.

Security Assertion Markup Language (SAML) provides a technology neutral way to exchange security information using XML to communicate authentication, authorization and other user attribute information. SAML also allows interoperation across different platforms such as J2EE, .NET and CORBA. WS-Security supports the use of a SAML token in the SOAP header.

The distributed authentication process implemented in the RLS prototype is shown in more detail in Figure 16.



Patient Lookup

1. User logs in / enters patient lookup query (demographics)
   a. authenticated against directory
   b. access logged
2. Request for patient record locations in SOAP envelope with user identity / roles over
   c. sender -side certificate used to sign message and receiver certificate used to establish SSL /TLS connection
3. Matching patient record locations looked up
   d. remote system authenticated against registry
   b. access logged
4. Matching records from CMPI returned
   c. sender -side certificate used to sign message and receiver certificate used to establish SSL /TLS connection
5. Patient record locations displayed for user selection

Medical Records Retrieval

6. Patient clinical records query entered
7. Request for patient medical records request in SOAP envelope with user identity / roles , and server key
   d. remote system authenticated against registry
   b. access logged
8. Patient clinical records retrieved
9. Clinical records returned to user
   c. sender-side certificate used to sign message and receiver certificate used to establish SSL /TLS connection
10. Clinical records aggregated and displayed to user

Figure 16: Authentication Mechanisms for Patient Lookup and Medical Records Retrieval

5.6 Messaging Patterns

RLS defines contracts to govern message exchanges that implement services. These message exchange patterns, or message scenarios, are the basic transactions that Web services are designed around. The logical components and services described in the previous section may be considered as the 'plumbing' (or the Common Framework) for the health information exchange. The framework is capable of supporting various message scenarios that support multiple use cases.

There are four common messaging interaction patterns that characterize service oriented scenarios:

- One-way: or fire-and-forget messaging, involves the sending of a message from requester to provider with no acknowledgement expected
- Request / Response: implies that a response message is generated for every request received by the provider
- Notification: may be considered a mirror image of the one-way pattern, where the provider sends a one-way message to the requester
- Solicit response: is the reverse of request / response in that the service provider sends a solicitation for a request to a requester

The contract between service provider and requestor is defined using a standard XML based language called Web Services Description Language (WSDL). Note that the WSDL 1.1 messaging terminology above have been superseded in the WSDL 2.0 specifications with more precise names; these do not have a material impact on the RLS specifications and are not used.[‡‡‡‡]

The message scenarios supported by RLS prototype are listed below:

Table 2 List of Messaging Interactions supported by RLS Prototype

#	*Name*	*Triggering Event*	*Interaction Type*	*Sender*	*Receiver*	*Receiver Responsibility*
1	Lookup patient locations	Practitioner receives consent from patient to retrieve medical history	Request / Response	Practitioner (via CDX Gateway)	RLS	Search master patient index Match patients using linking algorithm Return list of patient locations (clinical systems) and MRN
2	Publish patient index	Registration of new patient into Clinical System Patient consent is pre-requisite	Notification	Clinical System (via CDX Gateway)	RLS	Patient basic demographics and MRN (as maintained in the Clinical System) used to update CMPI.
3	Message logging	Passing of message through gateway	One way	CDX Gateway	RLS	Insert log message into standard format logging database

[‡‡‡‡] Web Service Addressing 1.0 WSDL Binding W3C Working Draft 15 February 2005, http://www.w3.org/TR/2005/WD-ws-addr-wsdl-20050215

#	*Name*	*Triggering Event*	*Interaction Type*	*Sender*	*Receiver*	*Receiver Responsibility*
4	Exception logging	Error condition in message processing	One way	CDX Gateway	RLS	Insert error message into standard format logging database and notify Administrator
5	Retrieve medication history records:	Authorized practitioner submits request. Patient consent is pre-requisite	Request / Response	Practitioner (via CDX Gateway)	Clinical System via CDX Gateway	Return requested medication history list in a standard message format

Each message scenario assumes that an error message (SOAP Fault) is returned by the receiver to the sender if the message results cannot be parsed or results in any application error.

The flow of logic across the different service layers in the RLS / CDX Gateway solution architectures is shown as sequence diagrams in Figure 17 and Figure 18 where the application process logic may be visualized as a series of messages exchanged between application services. The services oriented architecture is realized through implementation of messaging between application components, as shown here. Figure 17 depicts the interactions between RLS application components to provide the patient lookup service to users logging in to remote gateways.

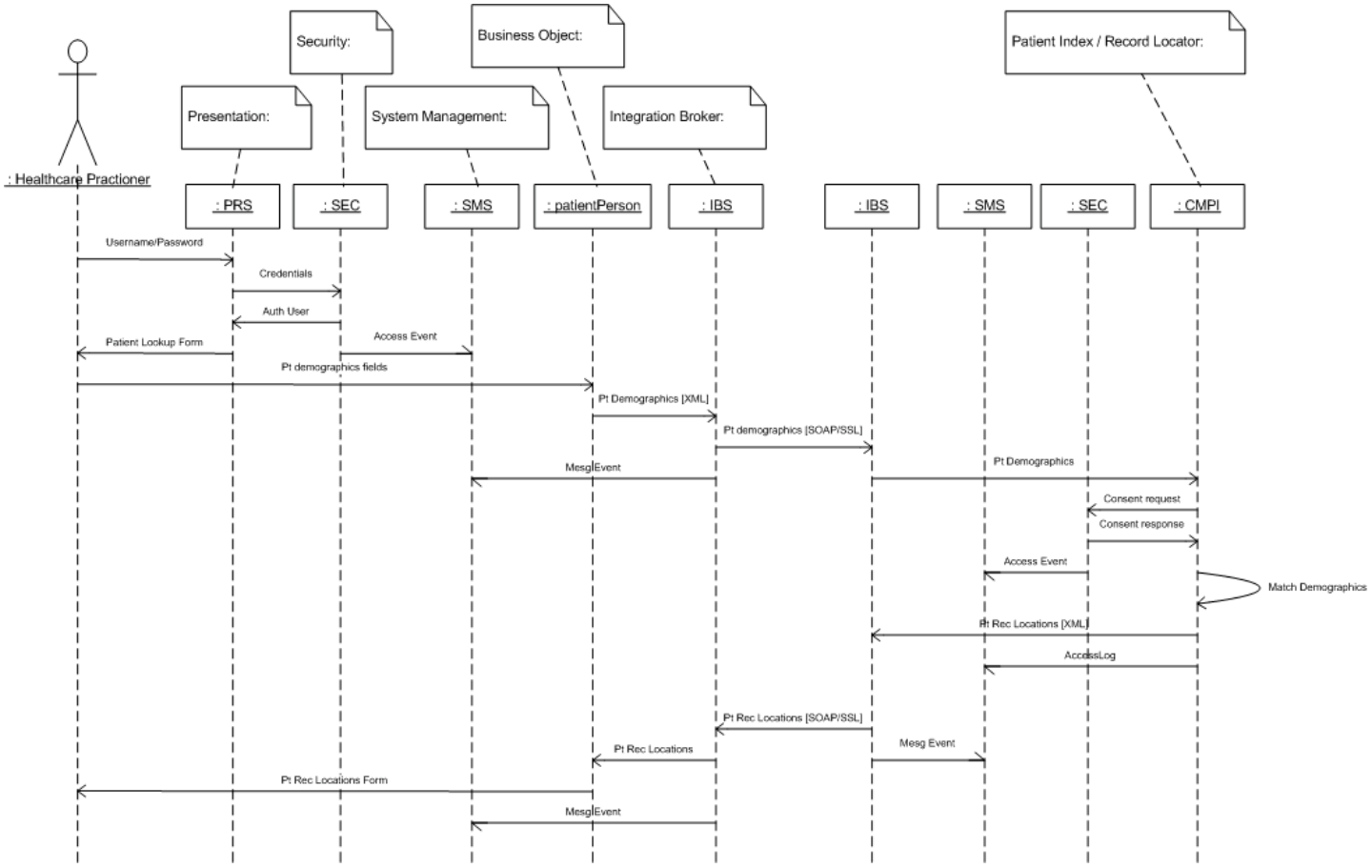


Figure 17 Patient lookup sequence diagram

As can be see in the figure, the Gateway services communicate with each other through the integration broker services.

The other core RLS service is that of accepting updates to patient indices from clinical data sources and applying them to the RLS CMPI. This may be visualized in the sequence diagram in Figure 18.

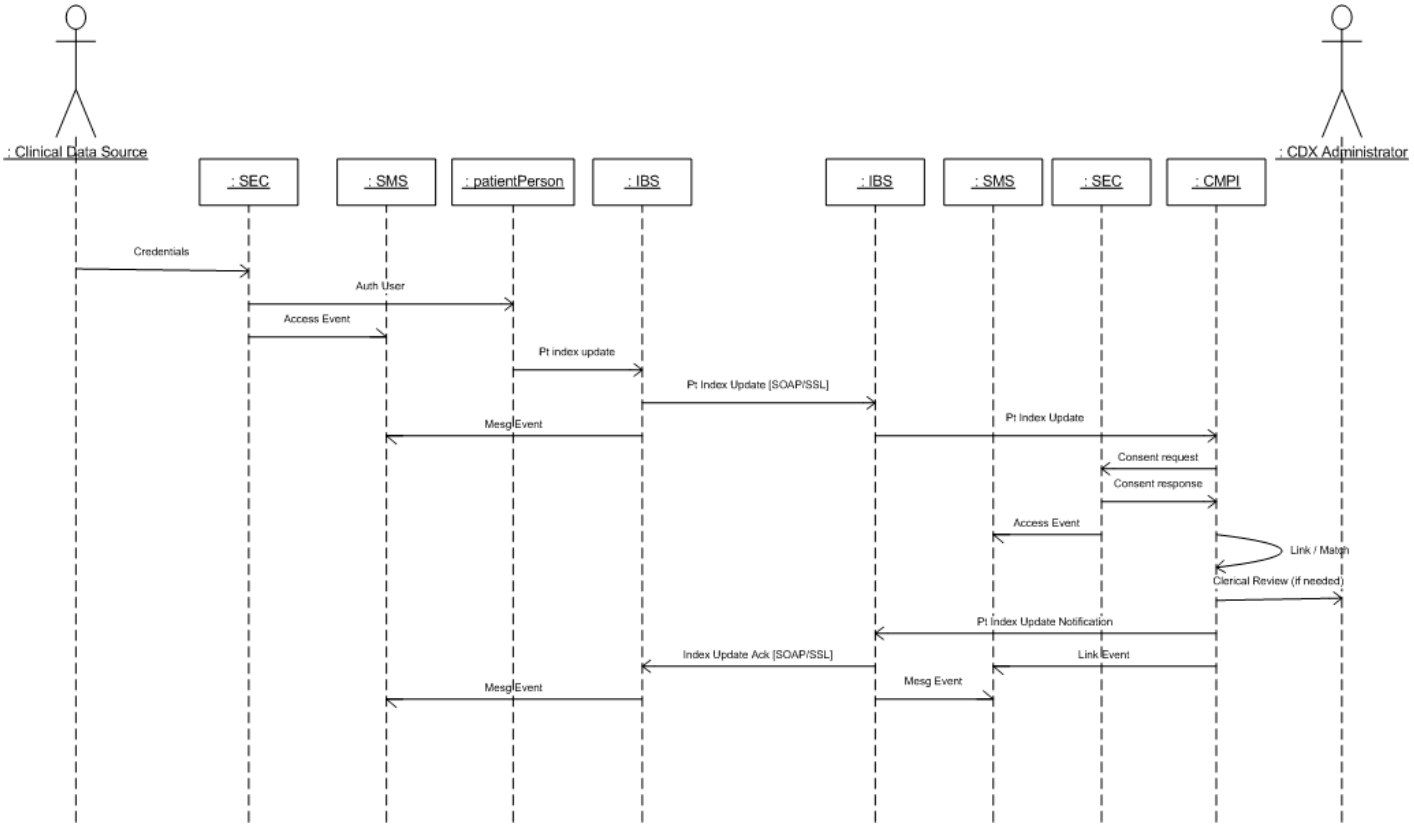


Figure 18 Publish patient index sequence diagram

6 Implementation View

This section describes how the relevant components of the RLS logical architecture are implemented. While the logical and process views provide more conceptual models of RLS, the implementation and deployment views relate more to the physical artifacts that make up the RLS-based interoperability framework. The implementation view presents a more detailed view of the organization of the static software elements of the RLS prototype than was presented in the logical view.

Note that implementation of systems based on the RLS architecture is to be undertaken by individual RHIOs, which have significant latitude of action. Each RHIO may choose implementation strategies based on their internal development methods and platform preferences. This section provides only generic implementation related information, and refers to the RLS prototype implementation architecture and platform for purely illustrative purposes.

RLS architecture is generic and platform agnostic, with interoperability based on open connectivity standards such as Web services, and semantic data standards such as the HL7 Reference Information Model. This section also outlines the relevant standards prescribed for an RLS implementation. Given that interoperability is critically dependent on the communication and data standards adopted, the expectation is that each implementation adheres closely to the recommended standards and specifications. More detailed guidance on implementing message format standards and specifications is provided in a separate document: RLS Messaging Communication Implementation Guide.

6.1 Overview

RLS is a classic n-tiered database application with a web-browser based user interface, and capability to interoperate with remote systems using open standard messaging over the Internet. The key interoperability function is realized through implementing a gateway service at each node in the network that provides essential presentation, business and data services and the ability to communicate with other gateways using web services and domain specific data standards. The RLS application may be viewed as comprising two large grained components (or service compositions).

- Patient Index service maintains and enables access to the community Master Patient Index (CMPI) and maintains a community directory / registry for the various clinical data sources, data and message standards, etc.

- CDX Gateways provides a 'web-service' wrapper and other utility services to support message based interoperation for the RLS as well as for each clinical data source in the network.

The patient index service architecture essentially comprises components that provide Patient Record Linking / Matching and Patient Record Search services. The CMPI database contains basic patient demographic information and patient identifiers as maintained in the various clinical data sources that publish into the CMPI. Each patient record is tagged with the network resolvable address of its source.

The unified patient view is achieved through linking / matching of patient records based on demographic attributes. Record matching may use deterministic (exact) matching of patient demographic attributes or a probabilistic algorithm which takes into account the variations in source data due to data entry anomalies. The architecture implementation supports the swapping of matching algorithms by change of run-time parameters.

CDX Gateways enable the interoperation of backend legacy clinical systems with each other and with the RLS through exchange of messages (of various formats) with other gateways using Web services protocols over HTTP transport secured with SSL/TLS (or HTTPS).

6.2 Components and Layers

A logical view of the RLS application architecture was presented in Section 4.

A more detailed breakdown of the components of the RLS, their logical groupings (layers), and implementation platforms used in the prototype project is shown in Figure 19

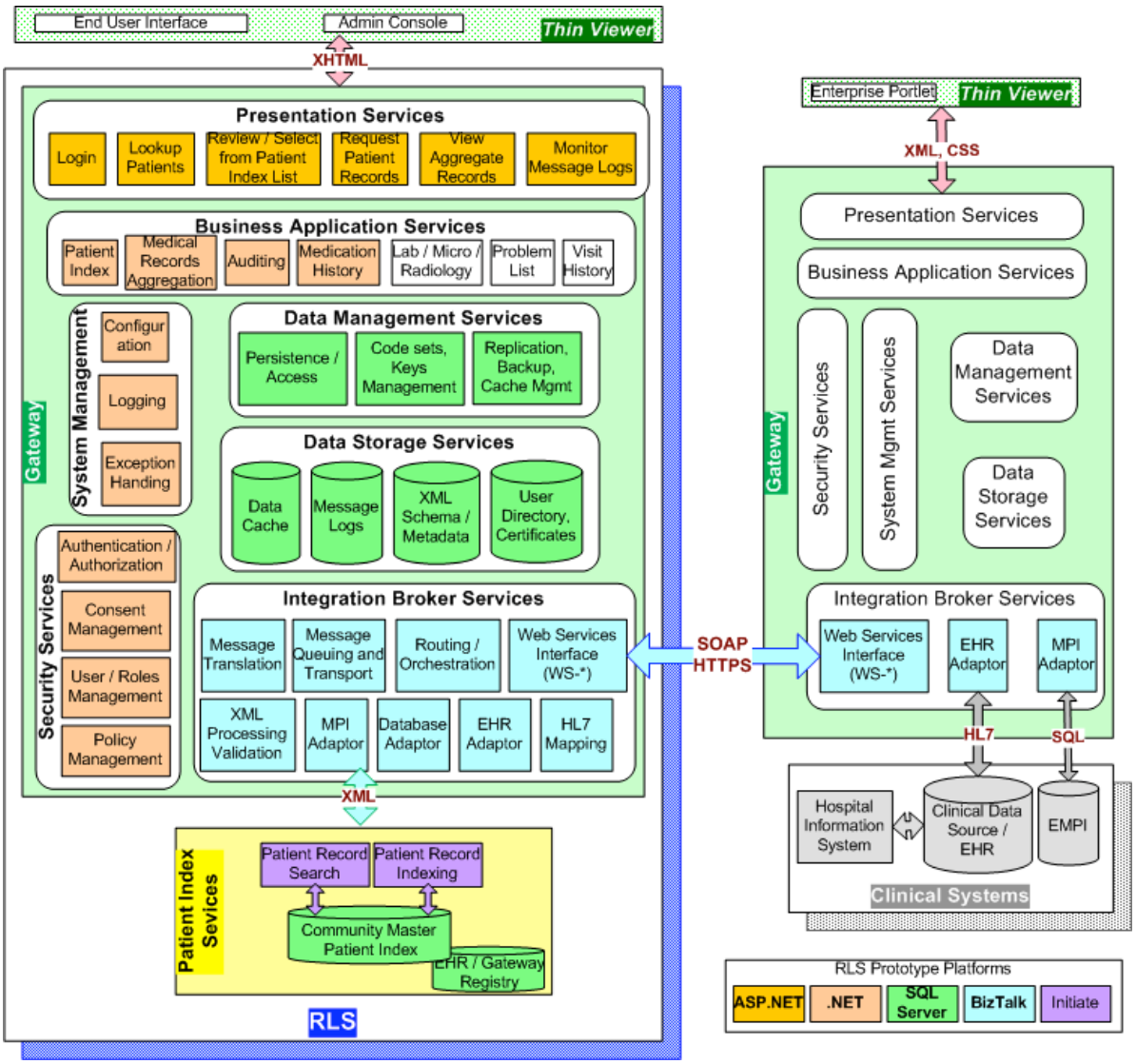


Figure 19 RLS and CDX Gateway components and sample (prototype) implementation platforms

The diagram shows one EHR node (comprising a clinical system and a gateway) connected to the RLS. One may imagine many such nodes also communicating to the RLS through the Web service interface component in the Integration Broker service of the gateway. CDX Gateways interface with the local clinical systems through the EHR adaptor. A design objective would be to isolate and layer services such that only the adaptor components in the Integration Broker service layer would need to be customized for each site where the Gateway is deployed.

The service layers are independently deployable units, which are orchestrated by a service broker or bus. For example, the security and systems management layers are shared across multiple business application services. Thus, all the service layers are loosely-coupled and reusable. Implementations of RLS may vary based on the specific needs of the site, with the potential for sites to tighten coupling as necessary. They may also use alternate infrastructure services (e.g. security) that conform to local standards as long as they also honor the interface standards used for RLS.

More details of the functionality of each of the components that comprise the service layers are provided in Table 3.

Table 3 CDX Gateway Service / Components Description

Service / Components	*Description*
Presentation Services	Formats data display to meet end user interaction and display device requirements
Login	Enter username and password to gain access to RLS
Lookup Patient	Patient demographics data entry
Review / Select from Patient Index list	Present list of potential matches of patients to demographics data entered, to be selected from
Request Patient Records	Selection of specific patient medical records to be retrieved from source
View Aggregate Records	Present patient medical records received from multiple sources in common format
Monitor Messages	View log of messages passing through RLS
Manage Access Control Policies	Create and maintain access control policies
Manage User Identities	Create and maintain user identity information and roles to which assigned
Business Application Services	Key functional components that house business rules and execute business logic on clinical data to render it comprehensible to the healthcare practitioner
Patient index	Patient index business object
Medical records aggregation	Application object that merges medical records received from multiple clinical data sources
Medical records request	Standard application object that mediates the data request entered by users on a screen and the data access services
Auditing Services	Support the auditing of access to medical data using logs of all significant events in gateway operations
Medication list	Meds administered business object
Lab / Micro / Radiology	Laboratory, microbiology, radiology results business object
Images and Waveforms	EKG/EEG graphs, Radiology imaging etc.

Service / Components	*Description*
Notes / Reports	Clinical documentation business object
Visit History	Patient encounters business object
Problem list	List of current diagnoses business object
Data Management Services	Manage application access to data storage and processing of data in data storage layer. Isolates the business layer from the details of the data storage service. Supports management of metadata about data stores and repositories in system
Persistence / Data Access	Provide standard data access to business application services that is independent of the underlying data storage technology or database management systems
Code sets and Key management	Manage disparate code sets and generated keys for data imported from multiple systems
Replication, backup, data cache management	Technical data management services. Support data aggregation and asynchronous data streams management Data cache for clinical data (where required).
Data Storage Services	Provide reliable, secure data storage for efficient access by data management services
Data Cache	For storing temporary copies of data retrieved from clinical data sources for faster response to user queries. This could be a significant component for pilot / production implementation since it buffers the clinical data sources from external queries. However, there are significant business and technical issues to be resolved for clinical data caching.
Message store, Logs and Audit	Persistence mechanism for reliable messaging and for monitoring of message flow (may be part of System Management Services)
User directory, Certificates store	Data store for security services layer (may be part of Security Services)
XML Schema and metadata services	Repository of message and data standards for reference – both run time and design time
Integration Broker Services	Manages flow of messages through the Gateway that serves RLS and the clinical systems that communicate with RLS
Message Queuing and Transport	Provides store-and-forward capability and manages connection to messaging infrastructure. Supports asynchronous messaging between loosely coupled services
Message translation	Transform message formats based on mappings
Routing / Orchestration	Routes messages to the appropriate destination channels
Web Services Interface	SOAP and WSDL processing along with other WS-* service implementations, e.g. reliable messaging, error handling, security
XML Processing	Serialization/deserialization of messages, validation of messages against XML schema, and translation from one schema to another using XSLT
HL7 Mapping	Conversion from HL 2.x messages to RLS standard formats (HL7 v3)
EHR Adaptor	EHR system specific component with ability to extract required medical records from EHR system
MPI Adaptor	Component to interface with Master Patient Indexes (including CMPI)
SQL / Replication / ETL Adaptor	Data movement from one data store to another, includes transformation as needed and typically executed in batch (bulk) mode
Systems Management	Besides application management functions shown below, extends over the long term to cover remote deployment

Service / Components	*Description*
Services	extends over the long term to cover remote deployment, configuration, administration and patch application of distributed Gateway from central location
Logging services	Interface for application to log processing events
Exception Handling Services	Interface to raise and manage errors in application processing
Configuration	Interface to manage the configuration of the RLS application. E.g. setting record linking /matching algorithm, logging levels etc.
Security Services	Manage the implementation of security to control access to the system and protect confidentiality and integrity of data in the system
User / Roles Management	Manage the creation, updates, and deletion of actors authorized to use RLS
Authentication / Authorization / Personalization	Validate that actors (user or system) is who/what they claim to be
Consent Management	Manage individual patients consent to let healthcare practitioners view their medical records
Policy Management	Provide interface to configure and manage rules for access to healthcare data, auditing, and secure operation of RLS

The services that make up the Patient Index Service of RLS are described in Table 4.

Table 4 RLS Components Description

Service / Component	*Description*
Patient Index Services	Maintain patient records sourced from multiple clinical systems and provide access to data
Community Master Patient Index	Multi-enterprise Patient Index data store maintained in the RLS
Patient Records Linking / Indexing	Identify multiple patient records pertaining to the same individual, but created with potentially different attributes
Patient Record Search	Search for index for patients matching the demographic attributes entered by healthcare practitioner

The interfaces between the application components shown are the subject of local implementation decisions. The general bias of SOA-based applications is to use XML messaging interfaces between components. The CDX Gateway integration broker service may be used as a 'service bus' that mediates connections between the coarse grained services that make up the SOA-based application. It should be recognized that a messaging interface often has a system performance overhead and that this penalty may not, in some implementations, be fully offset by the benefits of true loose coupling of application components. In such cases, implementations may choose to start with tighter RPC-style interfaces between application components, and migrate to SOA as performance management allows.

6.3 Implementation Topology Options

As shown in Section 5 there are several processing models that can be implemented with the gateway based network architecture. Each gateway is an n-tier application with distinct presentation, application, data storage and integration services. This allows the functionality and data storage at each gateway service to be varied based on the degree of centralization or decentralization desired.

Being based on the Internet, the health information network has a fully connected mesh topology at the transport (and connectivity) level. The application and data distribution across the network nodes determine whether an interaction between nodes is server-based or peer-to-peer. In server-based networks some computers (clients) consume services provided by others (servers). In a peer-to-peer network, the computers on the network can act both as clients and servers, and are referred to as peers. The RLS-based health information network is a hybrid, with some key services (record location) being provided centrally, while others (clinical data exchange) are consumed on a peer-to-peer basis.

RLS' service oriented application architecture (SOA) and the Web service based network supports multiple application and network configurations with varying degrees of data and application distribution. This derives from the fact that a loosely coupled, peer-to-peer model offers the ability to 'tighten' the coupling or centralize the architecture as needed by local implementations, whereas *a priori* centralization does not offer such flexibility. In effect the RLS-based network leverages the strengths of the Internet where a fully connected network allows varying service topologies to be used based on requirements.

The RLS-based network architecture seeks to find the right balance between being a potential single point of failure in the middle and reducing the processing footprint at the nodes. The proposed SOA model enables the intra-RLS service distribution to be adjusted and tuned for optimal performance at local, regional and national scale.

A case for data decentralization can also be built on patient privacy protection grounds. Recent security episodes and public perception suggest that the likelihood of data spills is reduced by not creating a large centralized repository of patient health information. Leaving protected health information in local clinical systems, and using a federated peer-to-peer clinical data exchange model reduces the likelihood of catastrophic data spills. Where local clinical systems are accessible from the network, the architecture anticipates data being cached by a hosted gateway service, which would serve as a proxy for the legacy clinical system (similar to an application service provider (ASP) model – a hybrid variation on centralized services).

An additional consideration is the messaging architecture for inter-RLS communication. Given that the health information network nodes are all Web-addressable each RLS sub-network node could connect to a remote RLS sub-network node on a peer-to-peer basis. However, as the security discussion indicates the need for each node to authenticate to each other impacts the scalability of such connectivity. An intermediary bridging service is required that can also be provided by the CDX Gateway at the RLS.

The different implementation options are listed in Table 5 below.

Table 5 Implementation Topology Options

Implementation Topology	*Description*	*Decision Criteria*
Peer-to-peer using gateways for transport mediation and aggregation	Clinical data are distributed and managed within their clinical systems Central patient registry for record location Gateways translate messages from local to network standard format Message and data aggregation at each gateway node	Service oriented architecture provides maximum flexibility in linking disparate systems No single point of failure (SPOF) No centralized command and control Increased mediation/ aggregation functionality at Gateway … complex distributed administration
Peer-to-peer with gateways maintaining clinical data caches	Gateways in addition to facilitating interconnectivity also serve as proxies for clinical data stores	Clinical data sources maintain autonomy Operational clinical systems are not subject to unpredictable query loads from network users Data replication needs to be set up and maintained between clinical database and proxy cache
Hub and spoke with distributed data sources using central mediation service for message routing	While clinical data remains at network nodes, messages are all routed through a central service Central service handles message and data aggregation	The gateway at the RLS could serve as the central routing and mediation service Lighter weight gateways at the edges minimize network joining overhead
Hub and spoke with central data repository	Variation of the distributed proxy data cache wherein data from clinical data sources are moved to central data repository Gateway function is purely transport mediation	Increased central data management and security overhead Reduced participation rates from clinical enterprises Very light-weight distributed gateway High degree of data conformity required

Implementation Topology	*Description*	*Decision Criteria*
Inter-RLS data sharing on a peer-to-peer basis	Use Gateway service at each clinical system to communicate across RLS sub-networks	Sharing of information across communities needs to be independent of data distribution with the RLS sub-network Trust relationships need to be built on very large scale across all nodes in all sub-networks
Inter-RLS data sharing using a 'central' intermediary service	Use the RLS Gateway to provide intermediary services to mediate patient lookup and clinical data exchange between different sub-networks	Trust relationships need exist only within an RLS sub-network and between RLSs

Centralization and distribution are relative concepts and network topologies typically exist somewhere on a continuum between the two. The RLS architecture principle of federated data and centralized directories is currently considered best practice, but may well need to adapt to different models as technologies evolve. The above list of possible implementation options shows that the proposed architecture is flexible and adaptive.

6.4 Security Model

The RLS architecture principles recommend a delegated authentication model as the most practical approach to achieving the rigorous security and privacy demands on a health information network. Users are authenticated at the gateway service that they use to connect to RLS. Each gateway service is a full member of the clinical enterprise trust domain where it is deployed. Users wishing to access the health information network have their identity and authorization verified by enterprise security processes integrated with the gateway service.

Delegated enterprise security processes are expected to fully conform to HIPAA regulations and other clinical system and local, regional and national security requirements. Once authenticated, the user's identity is embedded in each message flowing through the network, and is logged for comprehensive audit-ability. In addition, the RLS security model calls for authentication of sender and receiver systems using SSL/TLS (or HTTPS) for all messaging interactions. The digital certificates required for SSL/TLS based client and server authentication should be issued by trusted third parties. X.509 certificate life cycle management is recognized as a significant overhead, and automated support is essential as the network expands.

This basic security model that all network nodes must adopt to use RLS is considered adequate for point-to-point (SOAP server to SOAP server) message confidentiality, authentication and integrity. More comprehensive network security covering intermediaries, application-to-application encryption, etc. would need to use message level security.

The RLS security model foresees the migration to WS-Security based authentication across gateway services using XML Digital Signatures and XML Encryption to address confidentiality, authentication, integrity and non-repudiation requirements. WS-Security standards are available for X.509 digital certificate based message signatures and encryption, but implementations are relatively immature. After stabilization of the RLS basic transport level security, implementations should migrate to message level security using WS-Security.

A fully federated architecture would require individual user credentials to be managed at each node, which would pose a significant identity management problem. While federated security standards have been proposed, these are currently not proven in large scale inter-enterprise networks of disparate systems. RLS architecture should, over time, evolve to federated authentication and authorization models using Liberty Identity Federation Framework (ID-FF) and the Secure Assertion Markup Language (SAML) as mature implementations become available.

6.5 Implementation Platforms

There are several platform options available to implement the open standards based RLS architecture. Following the principle of no proprietary technologies, this technical overview does not recommend any specific platform for RLS. As guidance for identifying the appropriate platform-specific tools for the various components of RLS, the discussion below covers the experiences of the prototype development project.

The RLS prototype is developed on the Microsoft .NET platform for local reasons relating to skills and resource availability. The platform choice is based on practical considerations that apply to only RLS prototype development. Other technologies could as well be used, and it is expected that future implementations of RLS are based on other platforms. The choices made for the prototype components and possible alternatives are provided in Table 6.

Table 6 Prototype Platform and Options

Service Layer	*Prototype Platform*	*Alternatives*
Presentation Services	ASP.NET	* JSP * PHP

Service Layer	*Prototype Platform*	*Alternatives*
Business Application Services	.NET components	* Java Servlets * EJB Session Beans * PHP / Python / Perl
Data Management Services	ADO.NET using .NET framework services	* EJB Entity Beans * Java Servlets * PHP / Python / Perl
Data Storage Services	Microsoft SQL Server 2000	* Oracle DBMS * IBM DB2 * MySQL * PostgreSQL
Integration Broker Services	Microsoft BizTalk Server 2004	* BEA WebLogic Integrator * IBM WebSphere / Mercator * InterSystems Ensemble * Orion Symphonia * SeeBeyond eGate * Combination of Enterprise Service Bus (Sonic MQ) and XML utilities (Altova XML Suite)
Adaptor Services	Custom components built on BizTalk framework	* Packaged adaptors from Integration broker vendors above
Messaging Services	Microsoft BizTalk Server 2004, which uses MSMQ	* IBM WebSphere MQ
Systems Management Services	Custom .NET components using .NET framework	* CA Unicenter * IBM Tivoli * Microsoft Management Services
Security Services	Custom .NET components using simple database table for user identities / credentials	* Novell Odyssey * Sun ONE * CA eTrust

6.6 Interconnectivity and Data Standards

Standards play a central role in the interoperability framework that RLS is part of. Policies and data standards may be considered the two pillars of the healthcare information network interoperability architecture. Technical standards that underpin the RLS-based common framework are described here.

While the healthcare industry in the US has no shortage of data exchange standards, clinical systems interoperability remains a major challenge. The problem is more one of choosing from several candidate offerings from various standards development organizations, and specifying coherent interoperability profiles that are easy to implement.

Interoperability standards can be specified across technology, data, application and organizational domains. Given the restricted problem domain of RLS, the common framework focuses on standards that are directly relevant to the use cases within RLS' immediate scope. To advance decoupled development of interoperable systems and rapid adoption of the data sharing architectures, the RLS specification seeks to cover a minimum set of standards rather than make "all or nothing" recommendations.

At a high level, system interoperability standards may classified under the following categories:

- Domain Data Content and Structure Standards: includes information models, data naming standards, and controlled vocabularies. These represent semantic specifications that support business process level interoperability
- Messaging and Transport Standards: covering message packaging, transport and network protocols. These may be considered more in the realm of syntactic standards that support technical interoperability

As implied by its name various domain standards exist for the different clinical domains of data to be shared. However, it is possible to standardize on a common messaging and transport protocol that can be used across all the business domains. The technical standards decision is seen as relatively less contentious and will be discussed first.

6.6.1 Messaging and Transport Standards

Given the architectural principle to use open standards and the Internet for connectivity, the RLS uses Web services as the transport layer standard. This determination drives a range of other standards, which may be represented in the form of a technology stack. A common view of the 'Web services stack' is shown in Figure 20.[§§§§]

[§§§§] Web Service: Program Integration across Application and Organization boundaries, Tim Berners-Lee, http://www.w3.org/DesignIssues/WebServices.html, 2003-07-24

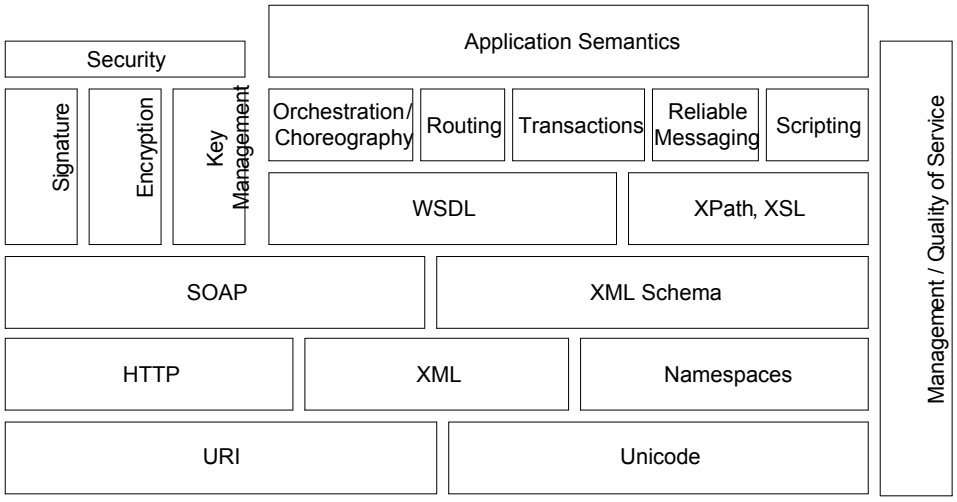


Figure 20 Web services stack

At the base of the stack are the HTTP and URI standards. The World Wide Web is evidence of the massive scale interoperability engendered by just these two enabling standards. Web services, based on SOAP (message packaging) and WSDL (message exchange contract) format standards, which in turn use XML and XML Schema as message notation and description standards, leverage transport layer interconnectivity to connect the data and application layers. Other technologies that are in wide-spread use and integrated in XML based messaging are XML Namespaces, XPath, and XSL. The other functional boxes in the stack have associated standards as well, but these do not have as high a degree of industry consensus about them.

Given the varying stages of approval and acceptance of the various standards in the stack, RLS needs to focus on the essential protocols that support interconnectivity while presenting the lowest adoption overhead to network participants. The Web Services Interoperability (WS-I) Organization provides a profile that focuses on the core Web services specs such as WSDL and SOAP, and addresses known interoperability issues. More specifically, the 'Basic Profile' provides specific implementation guidance on the core Web services standards that should be used together to develop interoperable Web services. Implementers thereby have higher confidence on achieving interoperability using Web services products from different vendors. The WS-I Basic Profile 1.1[*****] offers the best choice for a candidate stack that RLS should adopt, and track as it evolves with the national (and global) technology environment.

[*****] WS-I Basic Profile Version 1.1 Final Material 2004-08-24, http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

In addition, WS-I has published a draft Basic Security Profile that should be used in implementing WS-Security based services.[†††††] The advantages of using the WS-I profiles include reuse of tools, implementation guides, and reduced costs, complexity and risks. A more restricted Web services stack with specifications that conform to the WS-I Basic Profile, is shown in Figure 21.

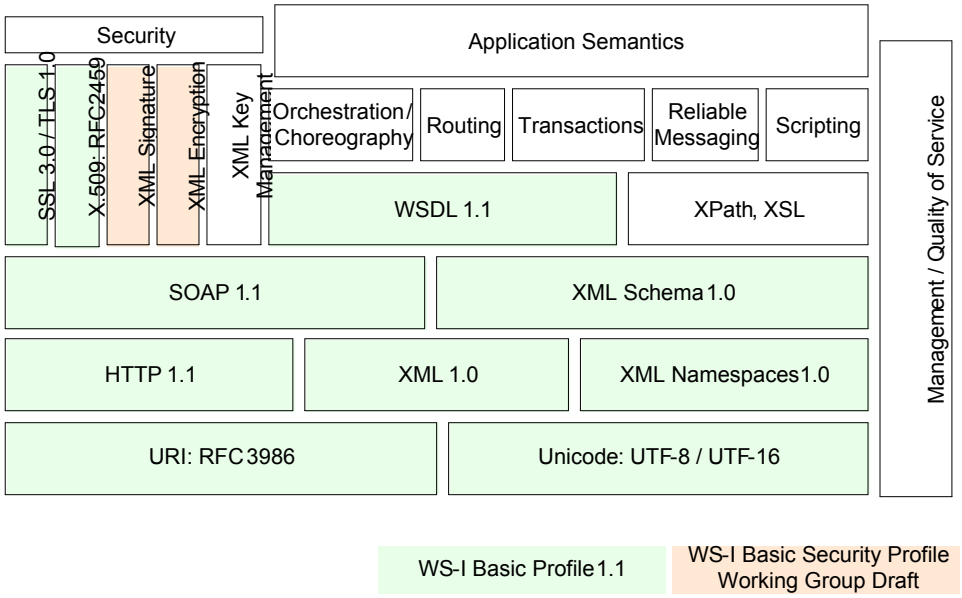


Figure 21 WS-I Basic Profile Web Services stack

RLS uses the WS-I Basic Profile as its core messaging and transport services standards suite. Web services specifications providing standard XML grammars for the other functions in the technology stack are growing and some (e.g. BPEL for process orchestration) are strong candidates to become mainstream standards in the near term.

As RLS grows functionally the appropriate specifications should be reviewed and incorporated into the standards stack. Ideally, this evolution of RLS standards should leverage profiles developed by WS-I, or other interoperability standards organizations. WS-I is currently engaged in updating the Basic Profile to include SOAP v1.2 and WSDL v2.0 which offer significant functionality improvements over the current versions in the profile. RLS implementations should develop migration strategies to SOAP v1.2 and WSDL v2.0, so that the added benefits from the new features can be availed.

[†††††] WS-I Basic Security Profile Version 1.0 Working Group Draft 2005-08-29, http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2005-08-29.html

An alternate XML based business to business messaging frameworks with strong claims to being 'industry standard' is OASYS' Electronic Business using eXtensible Markup Language (ebXML) framework[‡‡‡‡‡]. ebXML has been adopted by, among other, the U.S. Department of Defense for EMall, Center for Disease Control and Prevention (CDC) Public Health Information Network (PHIN), and NHS (UK) National Program for Information Technology (NPfIT). While the corresponding WS-* standards are maturing, as of this writing ebXML is clearly ahead in terms of stable specifications for reliable messaging, security, and exception handling. However, the rate of uptake of WS-* across the US market is higher than ebXML, particularly among applications and tools vendors. The perception that ebXML carries major implementation overhead has inhibited its use particularly among smaller organizations.

There are significant commonalities between the standards suggested for RLS and the PHIN ebXML stack. ebXML wraps another envelope on a SOAP message, and there is overlap between WSDL and ebXML's CPPA, as well as between UDDI and the ebXML registry. The RLS architecture is extensible to support ebXML based messaging, through extension of the Gateway Integration Services layer to include an ebMS type messaging adaptor.

6.6.2 Domain Data Standards

Having fixed on the Web services stack as its data transport standards RLS offers significant flexibility in choice of domain data standards. The primary use cases that RLS supports deal with publishing and looking up patient demographic information. The leading information model standard for patient information is the HL7 Reference Information Model (RIM), which is the basis for the new HL7 v3 message formats definition. RLS uses the HL7 RIM as the basis for data standards. RLS adopts a HL7 v3 message format for the various interactions it supports. Given the prevalence of HL7 v2.x messaging in the healthcare industry in the US, RLS also supports a 2.4 (XML) based message format. Details are provided in the RLS Communication Messaging Implementation Guide.

In general information exchange between nodes in the healthcare network may be visualized as occurring over a multi-layered set of standards, as shown in Figure 22.

[‡‡‡‡‡] ebXML OASIS http://www.ebxml.org/

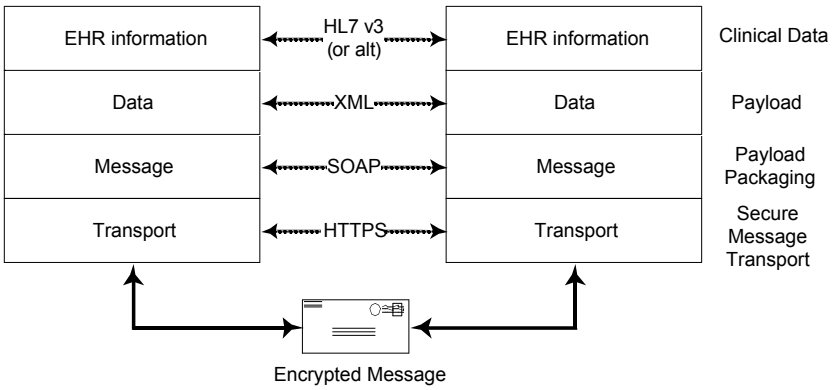


Figure 22 Interoperability Network Layers

EHR information in the RLS context refers specifically to patient demographic and identifier information. The same stack is applicable to general clinical data exchange in the healthcare information network. The domain data standards would be mostly drawn from the available HL7 format standards. However, legacy data formats need to be catered to such as NCPDP Script for prescription medication data, and DICOM for radiology imaging.

The interoperability framework is focused on messaging and data standards. This is in keeping with the SOA principle that interfaces trump implementation. The internal implementation details of the RLS Patient Index service or Gateway service are not relevant to the interoperability of the different network nodes. The two interfaces internal to a network node are:

- User access to the RLS is from standard Web-browser clients that invoke presentation services from the CDX Gateway servers. The user interface is generated in strict XHTML so that enterprises can use CSS and XSL style sheets to customize the user experience as necessary.
- Gateway services interface with the local clinical data systems through standards interfaces such as HL7 messaging, or SQL database queries. If the Gateway is to serve as a clinical data cache to offload queries from the transaction clinical system, then data feeds need to be built to move clinical data into the Gateway data cache.

6.6.3 Comprehensive Standards List

The full suite of standards covering support functions for RLS implementation and use in the healthcare information network is listed in Table 7.

Table 7 List of Standards

Component	Specification	Comments
Hypertext Transport	HTTP/1.1: RFC 3818	Base message transport layered on TCP/IP
Directory access	LDAP v3	
Domain name services	RFC 1035	
Transport security	SSL v3 / TLS 1.0	

Component	Specification	Comments
Encryption algorithm	3DES	
Message Hashing	SHA256	
Message Signing	RSA FIPS 186-2	
Web service message	SOAP v1.1	Upgrade to SOAP v1.2
Web services description	WSDL 1.1	Upgrade to WSDL 1.2
Web services basic interoperability profile	WS-I Basic Profile 1.1	
Web services choreography	BPEL4WS	
Web services security	WS-Security	
Web services addressing	WS-Addressing	
Data integration metadata / metalanguage	XML 1.0	Legacy formats (e.g. HL7 v2.x, NCPDP) do not all use XML.
Data integration metadata definition	XML Schema 1.0	Non-XML based messages do not have a standard schema notation
Data transformation	XSL: Extensible Stylesheet language, http://www.w3c.org/TR/xsl	
Data modeling language	UML	
Data model exchange	XMI	XML based metadata interchange
Message signatures	XML Signature	Signatures are embedded in the SOAP Header
Message encryption	XML Encryption	Encryption information is provided in SOAP Header
Registered namespaces	URI (Uniform Resource Identifier) http://www.w3.org/TR/2001/NOTE-uri-clarification-20010921/	URN: form of URI which uses a namespace for persistent object names
Scheme for site identification on the WWW	URL (Uniform Resource Locator): address of a resource which is retrievable using the Internet. http://www.w3.org/TR/2001/NOTE-uri-clarification-20010921/	
Identifiers using ASN.1	Object Identifier (OIDs) http://www.iso.ch/iso/en/ISOOnline.frontpage	
Scripting	ECMA 262 Script http://www.ecma-international.org/publications/standards/ECMA-262.HTM	
Domain data	Health Level Seven (HL7) v3	

Component	Specification	Comments
Clinical Terminology	SNOMED Sponsor: NLM (sourced from College of American Pathologists)	Clinical Terms creates a single unified terminology to underpin the development of the integrated electronic patient record by providing an essential building block for a common computerized language for use across the world

7 Deployment View

This section describes how the major components of the RLS logical architecture are distributed across hardware nodes in a health information network. Given that only the interface (inter-node messaging) specifications are expected to be consistent, deployment of systems based on the RLS architecture could vary widely based on local technology policies and preferences. This section provides deployment related information, for general guidance.

RLS components and services are designed to be flexible and highly configurable to enable deployment across a wide variety of sites. RLS requires the deployment of software on the following two types of server nodes:

- RLS Patient Index server which hosts the database of pointers to records in the clinical data sources, and routing information for each of the Gateways.
- CDX Gateway server which hosts the middleware that mediates data transfer from clinical data sources maintained at distributed locations (e.g. provider clinical systems, payer claims databases) and the Internet.

The RLS Patient Index is deployed at a central facility that provides robust data center management capabilities. This represents the one new patient data location that the health information network introduces. It is essential that HIPAA rules be observed at the RLS data center, much as they would in a covered entity.

Gateway servers are located within the circle of trust of the clinical system, and are typically deployed at the edge of the enterprise IT infrastructure zone in what is popularly known as the “DMZ”. An application firewall separates the Internet accessible gateway server from the internal network resources of the clinical enterprise. Only specific and authorized messages from the gateway are allowed past this application firewall.

Depending on volume of message traffic and throughput requirements, consideration may be given to deploying an ‘XML firewall’ that protects against malicious XML content based attacks. Such devices also feature XML processing capabilities, including XML Encryption and XML Digital Signatures using X.509 digital certificates based public keys. Typically, XML processing in hardware would provide significantly higher message handling performance. However, XML-aware network infrastructure is an emerging product category, and organizations should ensure that lock-in to proprietary features is avoided by insisting on conformance to open standards and verifying interoperability with XML software solutions.

Gateways can also invoke Web services on each other, enabling server processes to query the RLS. Such 'batch' mode queries and response processing will require additional functionality including the services of a job scheduler on the gateway server. The systems management services layer of the gateway is expected to manage these processes, which may be integrated with enterprise standard utilities as needed.

In a production network, deployment is expected to span a large number of CDX Gateway nodes that communicate with RLS and with each other. Sample production deployment topology is shown in Figure 23. Hardware sizing at each node is done based on production deployment requirements, which would be driven by network characteristics such as clinical transactions and patient registry volumes.

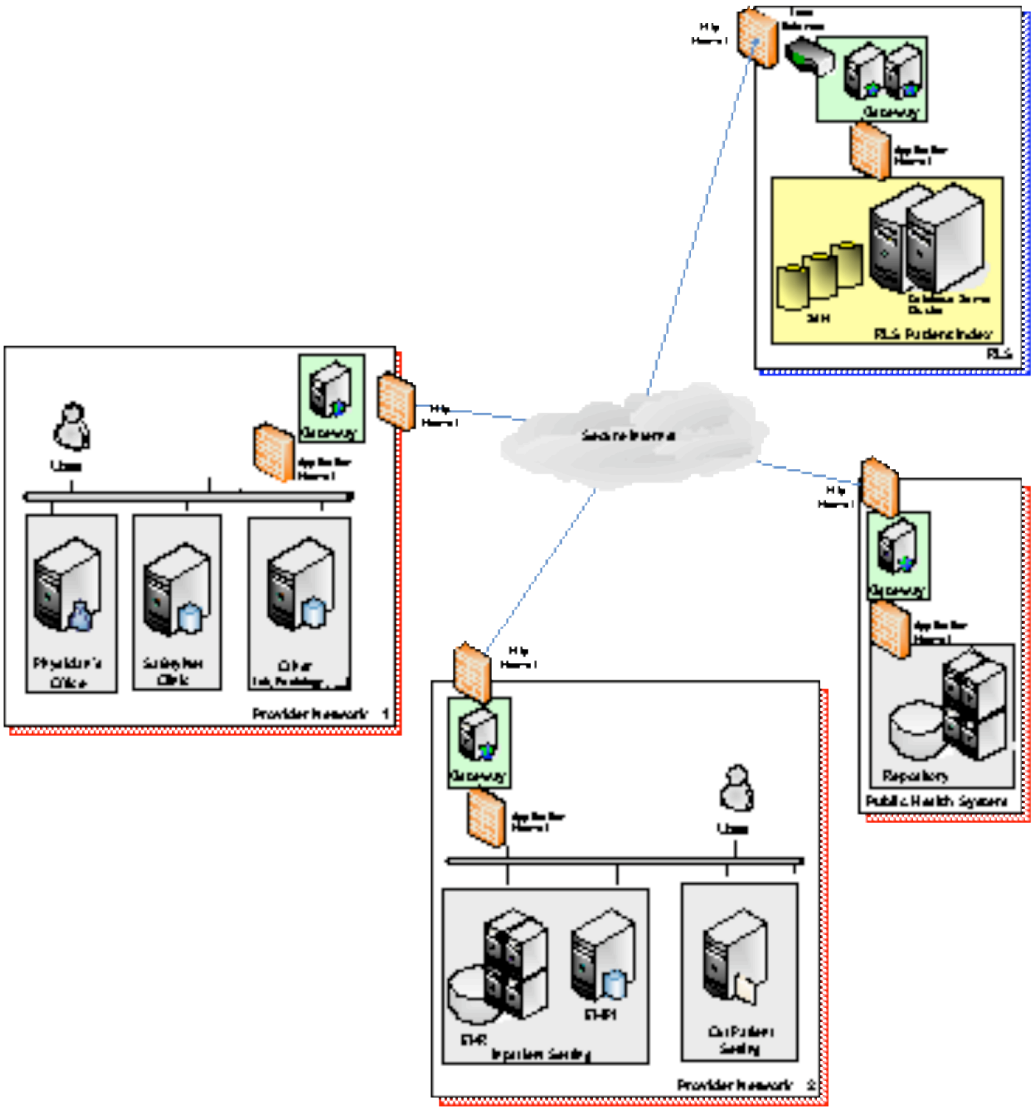


Figure 23 Potential Production Deployment View of RLS

7.1 Services Management

Systems management is critical to RLS-based network operations. The CDX Gateway architecture provides for comprehensive message logging and auditing capability. The same logs could be readily extended to support monitoring and reporting using simple scripting and system utilities. Several enterprise and network systems management tools exist that can be deployed to manage the CDX Gateway within the enterprise network context. Gateways need to be instrumented with the appropriate Simple Network Management Protocol (SNMP) agents for this purpose.

A Web services systems management standard: OASIS Web Services Distributed Management is expected to gain ground in the near future and advance SOA management using Web services. WSDM is based on SNMP and the Common Information Model and would represent a natural evolution from current distributed management standards to Web services based ones. The advantage of WSDM is that it uses Web service methods to manage distributed services such as those proposed for the RLS-based health information network, and therefore aligns well with the RLS architecture. Additional system management services would need to be added to the CDX Gateways to monitor and control message traffic using the WSDM protocol.

One of the key features of WSDM is SOAP based deployment of Web services. This would allow the Gateways to be deployed and configured from a central location (such as from RLS). This would further the goal of a utility service that can be cloned and deployed at the distributed network nodes with minimal disruption to the EHR systems at the node.

7.2 Security Services

Since RLS' major communication security functionality is embedded in the gateway, this offers a convenient approach to localizing the deployment and management of security services. The major security infrastructure that needs to be deployed with the RLS and CDX Gateways are the Digital Certificates required to support SSL and, in future, WS-Security.

Directory services (LDAP or Active Directory) are often used as certificate stores, and as the user identity and roles repository. Enterprises directories should be used if this shared service is available on the EHR network. Gateways should be interfaced with the enterprise directory using LDAP interfaces.

The reader is referred to directory and PKI documentation for detailed guidance on the deployment and management of public key certificates.

8 Data View

Following the federated data architecture principle, RLS persists minimal patient data centrally. The core of the RLS data store is a community Master Patient Index (CMPI) that supports lookup of patient electronic health record locations based on basic demographic attributes.

A canonical information model is used to develop reference XML schema that CDX Gateways use to send and receive messages based on the HL7 Reference Information Model[§§§§§] (RIM). A logical data model using standard Entity-Relationship diagramming notation is derived from the information model. All messaging services that RLS supports are integrated with the physical implementation of the logical model in the form of a relational database.

8.1 CMPI Information Model

The RLS information model view derived from the HL7 RIM is shown in Figure 24 for reference.

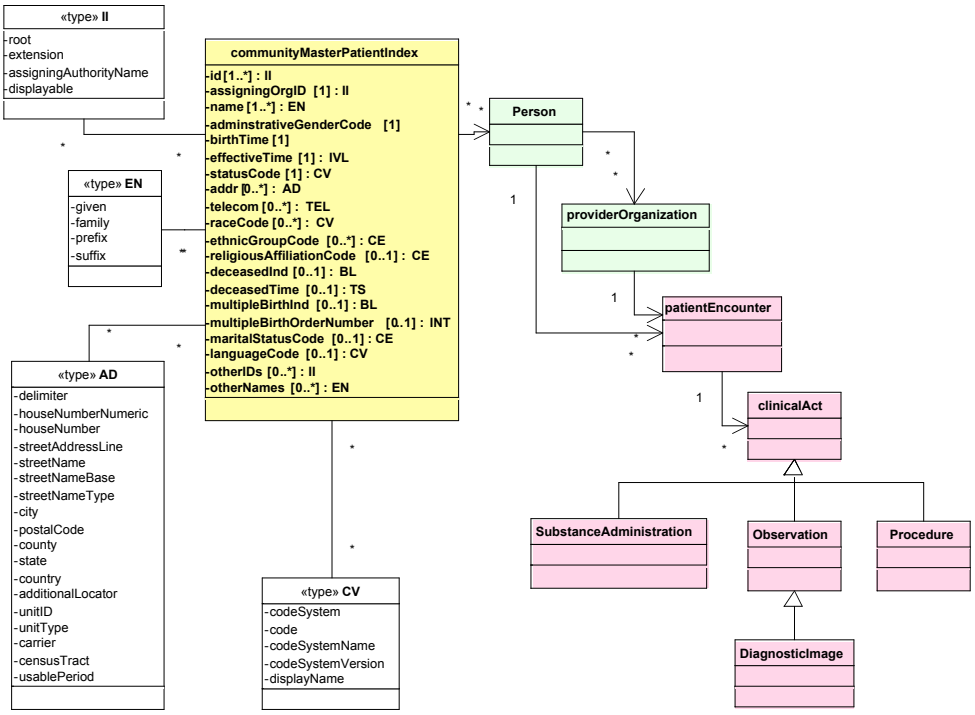


Figure 24 Information Model View

§§§§§ HL7 Reference Information Model, http://www.hl7.org/Library/data-model/RIM/modelpage_mem.htm, 2005-02-15

The community Master Patient Index follows the traditional MPI structure storing only 'pointers' to providers systems and patient identifiers therein, in addition to essential demographics attributes that can be searched on. The pointer to patient records is the *id* attribute of the communityMasterPatientIndex (CMPI) class. The list of attributes shown in the model view represents a set of all possible patient demographics. An RLS implementation would choose a specific subset of demographic attributes for the CMPI based on the specific community policies and requirements.

As can be seen from the information model, the patient EHR that the RLS index points to is a hierarchical abstraction of the RIM classes. With the patient index provided by the CMPI users can retrieve and select from visits or patient encounters at the provider facility. Users may then navigate from encounters to individual care records represented by the generalized clinical act class that may refer to procedures, observations (covering laboratory results, diagnostic images, etc.) and substanceAdministration (medication) lists.

8.2 Logical Data Model

The classes and attributes in the RLS information model are translated into entities and attributes of a logical data model that can be implemented physically in a relational (SQL) DBMS. The logical data model derived from the information model is shown in Figure 25.

The Entity-Relationship (ER) modeling notation used here is directly translatable to physical SQL databases. Entities have attributes corresponding to the class attributes of the information model. Attributes above the dividing line form the 'primary key' of the entity. Relationships between entities are denoted with lines that have a crows-foot notation to symbolize the 'many' end of a one-to-many relationship. These relationships result in the entity at the 'many' end inheriting the primary key of the entity at the 'one' end, as a foreign key marked as (FK).

The *identifiedPerson* entity represents the person as maintained in the source (EHR) system. The attributes of the CMPI entity shown do not signify the norm in any way. Select attributes from the *identifiedPerson* entity are replicated into the CMPI entity, based on community requirements for patient record matching. The identifying information (primary key) of patients in the *communityMasterPatientIndex* (CMPI) is formed by concatenating the identifiers of the *assigningOrganization* and the *identifiedPerson*. This combination of identifiers provides a unique key for the CMPI record.

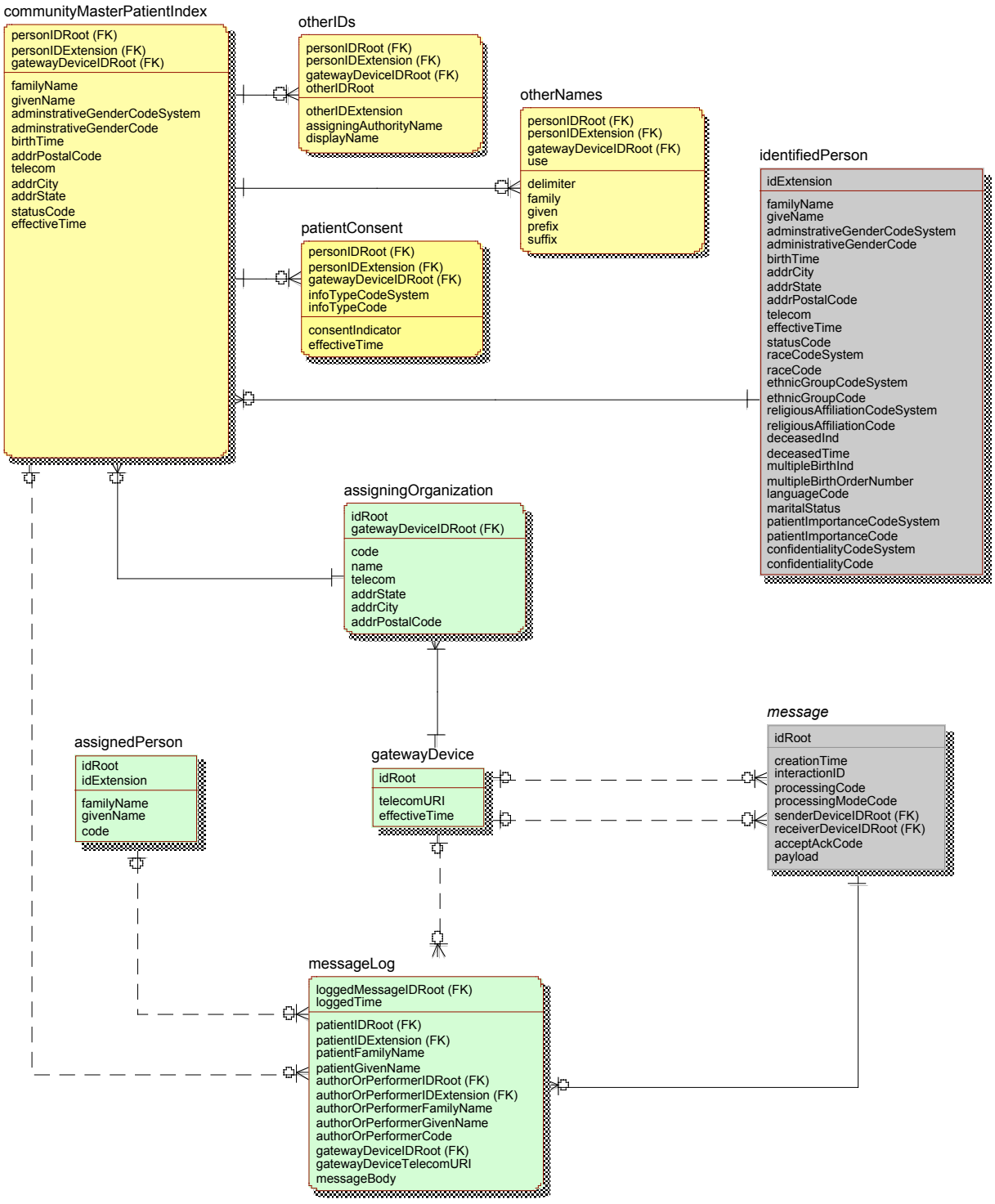


Figure 25 Logical Data Model

Within a clinical data source, patients are usually assigned *local* identifiers (e.g. MRN, chart number, etc.). In some instances alternate *standard* identifiers (e.g. Social Security Number, Medicaid numbers etc.) are used. Given the expected variability in data quality in diverse clinical systems, and the privacy constraints around some of the standard identifiers used (such as SSN) RLS does not distinguish between these two types of identifiers. Each is treated as a non-intelligent key to the patient record in the clinical data source.

The *gatewayDevice* entity is used to store the network address of the clinical data source managed by the *assigningOrganization*. Thus, along with the patient pointer information the CMPI returns the network address (of the Gateway) to which queries for patient medical records should be sent. When an EHR Gateway receives a patient medical data request it resolves the medical record location using the *personIDRoot* (*assigningOrganization*'s ID) part of the patient index, and redirects the query to the appropriate clinical data source.

The RLS supports use of multiple other identifiers for a patient such as identifiers used by ancillary systems. Theses additional identifiers are used more as attributes than identifiers, and may be used to search for the patient in the RLS. Standard identifiers, e.g. SSN, may be explicitly used as o*therID*s, if the RLS implementation policy and regulations permit. The *otherIDRoot* attribute entity represents organizations such as the Social Security Administration (for SSN) or state Registry of Motor Vehicles (for driver's licenses).

The *assignedPerson* entity represents the user who has access to the *gatewayDevice*. The user role that determines access rights of the user is carried in the '*code*' attribute (following the HL7 v3 implementation guide).

In addition to the business domain entities, messages and message logs are represented in the model. Messages are not stored physically in the RLS database except as XML strings in the message logs. Message logs are generic entities that may be used to store all messages that flow through the RLS/Gateway. This entity also carries patient and user attributes related to the message, which supports auditing of the logs.

8.2.1 Identifier Attributes

Translation from the object-oriented information model to classic relational data structures requires that the HL7 v3 data types be converted to SQL data types. The conversion is for the most part straight-forward where the components of the object attribute types such as II, EN, AD, etc. are flattened out to sequences of SQL data types.

Identifiers in the logical data model are formed from HL7 v3 instance identifiers (type II), and are worth examining in more detail since they are critical to understanding of the data returned by RLS. The II data type is defined as[******]:

> An identifier that uniquely identifies a thing or object. Examples are object identifier for HL7 RIM objects, medical record number, order id, service catalog item id, Vehicle Identification Number (VIN), etc. Instance identifiers are defined based on ISO object identifiers.

Instance identifiers are used for patient, organizations, devices etc. The HL7 v3 data type II has the following structure:

Element	*Description*
root	A unique identifier that guarantees the global uniqueness of the instance identifier. The root alone may be the entire instance identifier. This is number sequence that matches a pattern corresponding to DCE UUID, ISO OID, or strings consisting only of (US-ASCII) letters, digits and hyphens, where the first character must be a letter
extension	A character string as a unique identifier within the scope of the identifier root. If the root is used as a unique identifier, the extension is null
assigningAuthority	A human readable name or mnemonic for the assigning authority. Note: no automated processing must depend on the assigning authority name to be present in any form.
displayable	Specifies if the identifier is intended for human display and data entry (displayable = true) as opposed to pure machine interoperation (displayable = false).

The *extension*, *assigningAuthority*, and *displayable* attributes are all optional. The root may itself be used as a unique identifier, such as when it contains a UUID. In the RLS data model, the convention is to use UUIDs for transactional entities such as messages. Entities such as organizations and devices have fixed identifiers set up by the RLS, which may use OIDs or UUIDs. Patients have two part identifiers, where the *root* maps to the *assigningOrganization*'s id and the *extension* to the specific person id (e.g. MRN).

The *root* attribute of the patient identifier is set to the OID or UUID of the '*assigningOrganization*' that defines the id namespace (within which the id is unique). The *personIDRoot* of the CMPI is a foreign key mapping to the *assigningOrganization* primary key *idRoot* and the *personIDExtension* maps to the *identifiedPerson* primary key *idExtension*. The *personIDRoot* may be considered the prefix that RLS attaches to make the pointer unique in the CMPI.

[******] HL7 v3 Standard, Data Types Abstract Specification, September 2005

The primary key of the CMPI is not used for searching as the demographics attributes are. Searchable identifiers are stored in the o*therIDs* entity. For example, when a user specifies, where permitted, the SSN of the patient as a query criterion, the RLS derives the OID of the SSA using a lookup table of standard OIDs, which is then used to match the *patientIDRoot* of the *otherIDs* table and the given SSN is matched to the *patientIDExtension*.

8.3 Physical Data Model

The physical data model maps very closely to the logical model shown above. However, the physical tables are not all implemented in the same database instance since the architecture posits the RLS as a combination of a Patient Index service and a distributed Gateway service. The distribution of tables across the Patient Index and the Gateway is worth further discussion. The problem of OIDs management is also relevant to this design discussion.

The tables generated from the *CMPI*, *patientConsent*, o*therIDs* and o*therNames* entities reside in the Patient Index database. In addition the patient matching algorithm may create persistent secondary indexes to increase the performance of lookup queries. For example a probabilistic matching method may need to maintain a secondary index of Soundex transformed names. Since the RLS architecture needs to work with multiple matching algorithms, the patient matching component is treated as a separate service that maintains all the secondary indexes it needs. Optionally, the record matching algorithm may also generate a linking identifier that would be persisted along with the index. However, this would lead to increased maintenance overheads.

The remaining tables in the logical data model are created in the Gateway service data storage layer. Authorized user identity (*assignedPerson)* and *messageLog* tables are used for the purposes described above at Gateway services at each node in the healthcare information network, including the RLS node.

Organization and gateway information is maintained at each node based on the message processing requirements at the node. The RLS maintains the master list of Gateway services at all the network nodes. The Gateway service at the participating nodes maintains information on the various clinical data sources that it supports. The RLS does not require to know the details of the individual clinical data sources at each node. That information is abstracted by the Gateway service at that node. The RLS maintains a local copy of the OIDs for standard identifier assigning authorities as replicated from centrally maintained registries, e.g. the HL7 OID registry.

RLS accepts patient index data from the distributed sources with patient identifiers qualified with the *assigningOrganization* id. If these *assigningOrganization*s are not stored in the RLS Gateway, then the patient record in the CMPI is provided an additional prefix: the *gatewayDevice.idRoot*. The sequence of actions to build up the patient index in the CMPI is shown in Figure 26.

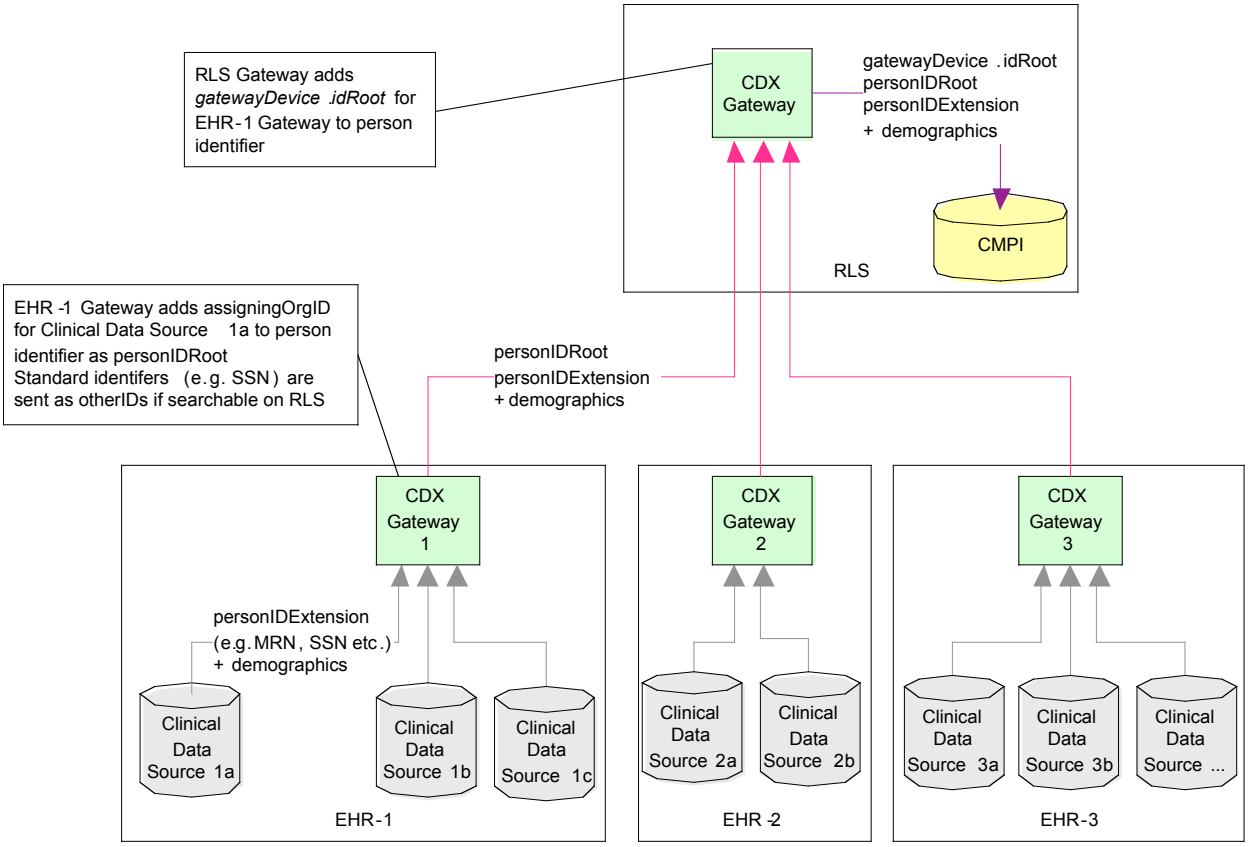


Figure 26 Patient Identifier Composition in CMPI

The patient record location provided by RLS in response to patient index lookup requests will contain the composite patient index made up of: *personIDRoot*, *personIDExtension* and the *gatewayDevice.telecomURI*. The recipient of this patient record location sends a query to the *gatewayDevice.telecomURI* with the composite *patientID* information. Since the remote Gateway maintains the *assigningOrganization* IDs it is able to resolve the composite patient identifier and retrieve the requested medical data.

As the above discussion shows, the division of labor between the Gateways at the clinical data source and the RLS requires that the appropriate cross-reference tables be maintained accordingly. At the RLS the various remote Gateways are assigned OIDs and maintained in the *gatewayDevice* table. The Gateway at the clinical data source, in turn, assigns OIDs to each clinical data source and maintains the cross-reference in the *assigningOrg* table. The use of OIDs is not mandatory; any local identifier system may be used as long as the patient identifier composition process described above is followed.

8.3.1 Data Quality Management

A general principle is that the CMPI database be a read-only version of patient records as they exist on source systems. The CMPI is intended purely for record matching and is not to be considered a patient registry. Data quality issues are expected to be resolved on the source systems, and cleansed data would replicate to the CMPI. The CMPI is therefore different from an Enterprise MPI in that no central data management organization is envisaged for an RLS. Data cleansing and quality services are not thought to be viable in a community of disparate, autonomous enterprises contributing data into the CMPI.

8.3.2 Data Cache

Message caching (logging) is likely to be required (over and beyond the persistence service offered by the MQ engine)

The data services layer in the CDX Gateway could serve as a cache for patient EHR data. Alternate architectures use clinical data repositories at the edge to serve the data requests received via information exchanges, so that core clinical data sources are not hit by these queries that could potentially impact the core clinical system performance. This aspect is discussed in more detail in Section 6.3. Some nodes may not want to push their EHR directly to the CMPI, instead may choose to expose their EMPI, or replicate their MPIs to the CMPI. Replicated MPI should have a 'time-to-live' based expiry, after which it must be refreshed.

9 Definitions, Acronyms, and Abbreviations

Term	*Description*
.NET Framework	The core programming model for using Windows as an application server. Offers native support for database connectivity and Web Services
Bus	Common conduit for message based communication between services
CDX	Clinical Data Exchange: A community utility that allows interchange of healthcare information between diverse medical systems
ebXML	Electronic Business XML: Standard messaging notation for business-to-business electronic business. Expected to supplant traditional ASC X12 EDI in future
EHR	Electronic Health Record: Clinical data collected in the course of delivering patient care, available in discrete digital form allowing access to individual data elements
FTP	File Transfer Protocol: Protocol for exchanging files over the Internet
HL7	'Health Level 7' (Refers to the seven layer network model popularized by ISO): Message format standards used for exchange of data between healthcare systems
HL7 RIM	HL7 Reference Information Model: Object model used in deriving new HL7 (Version 3) message formats
Interoperability	The ability of two or more systems (or components) to exchange information and to use the information that has been exchanged
LOINC	'Logical Observation Identifier Name Codes': Standard code set covering medical terms, procedures and diagnoses maintained by Regenstrief
Metadata	Data about data. Technical metadata describes how and when the data was collected, transformed and should be used. Business metadata provides the business meanings of the data
MIME	Multi-purpose Internet Mail Extensions: Standard format for non ASCII content sent over the Internet mail system
MPI	Master Patient Index (also called Master Person Index by some vendors): An electronic index that enables lookup of patient data distributed across multiple systems, to provide an aggregated view of patient's EHR
Prototype	A visual, functional model of the proposed software system. A prototype is developed for various reasons. The primary purposes of the RLS prototype are to validate software architecture concepts and to demonstrate the working of a software product to stakeholders.
Reference Implementation	A tool to demonstrate the practical feasibility of software standard specifications, or application programming interfaces (API).
RLS	Record Locator Service: An information service that locates patient records across systems that subscribe to the service
RxNORM	Clinical drug nomenclature produced by NLM, in consultation with FDA, VA, and the HL7 standards development organization. RxNorm provides standard names for clinical drugs and for dose forms as administered.

Term	*Description*
SAML	Security Assertions Markup Language: An XML framework for communicating security information (authentication, authorization, other attributes) between systems. SAML is independent of the security protocol used (e.g. PKI, LDAP, Kerberos, etc.) and promotes interoperability between disparate systems
Semantic Interoperability	Property of data exchange that ensures that the receiver of data understands what the sender 'meant' (contrast with mere 'syntactic' interoperability)
Service	Application system with a standard network callable interface.
Service Oriented Architecture	An application architecture comprising components, whose interface descriptions can be published, discovered and invoked. Components are said to be loosely coupled in that they have no knowledge of each other except for their respective interfaces and communicate with each other through messages W3C definition: A set of components which can be invoked, and whose interface descriptions can be published and discovered
S/MIME	Secure MIME: Version of the basic MIME protocol that supports encrypted messages based on RSA's public key encryption technology
SNOMED-CT	Systematized Nomenclature of Medicine – Clinical Terms: Standard code set covering medical terms, procedures and diagnoses maintained by College of American Pathologists
SOAP	Acronym, originally, for Simple Object Access Protocol; no longer considered an acronym. Lightweight XML based protocol for exchanging information in Web service based implementations. Primarily specifies the XML 'envelope' for a message.
SSL	Secure Sockets Layer: Protocol used to communicate private (encrypted) data over the Internet
UDDI	Universal Description, Discovery and Integration: Mechanism for web service providers to advertise services and consumers of services to locate them.
UML	Unified Modeling Language: general purpose language for specifying and visualizing software systems. Favored for object-oriented software development
URI	Universal Resource Identifier: The standard for naming and addressing resources on the Internet. The commonly known URL (Universal Resource Locator) is a form of URI
Vocabulary Domain	HL7 term for standardized set of values for coded attributes used in healthcare information messages; e.g. ObservationMethod, Race, VaccineType
Web Services	An application that is identified by an URI, and invoked via the Internet, using data exchange notations based on XML. By emphasizing simplicity and open standards, disparate applications can securely interoperate without knowing internal details of each other. W3C definition: "A Web service is a software system identified by a URI [RFC 2396], whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols"

Term	*Description*
WSDL	Web Services Description Language: standard XML based specification to describe the interface to a Web Service. Machine-interpretable standard form for describing the operation of a web service, represents the 'contract' that the Web Service honors with any requestor
WS-Security	Specification that encompasses all XML security standards related to SOAP messaging. Covers how security tokens are to be generated for SOAP message headers, how XML messages are signed and encrypted, etc.
XML	eXtensible Markup Language: Common notation used to represent data sent from one system to another. XML data files (or messages) use clear text and are 'self-describing' enabling human as well as machine understanding. E.g. HTML notation is based (loosely) on XML.