Service Functional Model Specification

Retrieve, Locate, and Update Service

Content Version 1.0.12
SFM Boilerplate Version 1.3

December 11, 2006

Project Lead	John Koisch, OCTL Consulting
Editor	John Koisch, OCTL Consulting
Authors	Ken Rubin, EDS Global Healthcare
	Alan P. Honey, Kaiser Permanente
	Scott Robinson, Kaiser Permanente
	Kensaku Kawamoto, Duke University Medical Center
	John Koisch, OCTL Consulting

Preface

Notes to Readers

This document is the Service Functional Model for the Retrieve, Locate, and Update Service, which is specified under the Service Development Framework process under the auspices of the Healthcare Services Specification Project (HSSP). Further context is given in the overview section below, but one key point to note is that the SFM provides a Service **Interface** specification, NOT the specification of a Service implementation. This is a critical distinction in terms of Service Oriented Architecture. There could be many different ways of implementing all or part of the functionality to support the behavior described in this specification.

Changes from Previous Release

This document relies heavily on the balloted RLUS specification that the HL7 organization reviewed in committee during the May, 2006 ballot cycle. While the content may be the same or similar, it has been reevaluated with respect to both member comments generated during balloting as well as ongoing HSSP work, especially in the Entity Identification and Infrastructure realms. See the HSSP wiki (http://hssp.wikispaces.com) for more information

Acknowledgements

RLUS represents a considerable undertaking by members of the Healthcare Service Specification Project. Many thanks to the members of the HSSP team who have contributed time and effort to this second revision. In particular, Alan Honey and Ken Kawamoto have spent considerable time ensuring that the end product meets the needs of a varied user community and have suffered interminable revisions. Jari Porrasmaa gave invaluable feedback in the May ballot cycle that greatly improved the effort. Ken Rubin facilitated, edited, revised, and focused RLUS with an eye on the larger strategic goals.

As usual, the HSSP team has proven to be a valuable, effective community to find solutions in this space. Thanks to everyone for another successful effort.

Table of Contents

1 Overview

1.1 Introduction and Scope

The Service Specification Development Framework Methodology is the methodology followed to define HSSP specifications. The methodology sets out an overall process, and also defines the responsibilities of the Service Functional Model (SFM). Section 2 sets out the business context for this particular specification, but firstly it is important to understand the overall context within which this specification is written, i.e. its purpose from a methodology standpoint.

1.1.1 HL7-OMG Healthcare Services Specification Project (HSSP)

The Healthcare Services Specification Project (HSSP) [http://hssp.wikispaces.com] is a joint endeavor between Health Level Seven (HL7) [http://www.hl7.org] and the Object Management Group (OMG) [http://www.omg.org]. The HSSP was chartered at the January 2005 HL7 meeting under the Electronic Health Records Technical Committee, and the project was subsequently validated by the Board of Directors of both organizations.

The HSSP has several objectives. These objectives include the following:

- To stimulate the adoption and use of standardized “plug-and-play” services by healthcare software product vendors
- To facilitate the development of a set of implementable interface standards supporting agreed-upon services specifications to form the basis for provider purchasing and procurement decisions.
- To complement and not conflict with existing HL7 work products and activities, leveraging content and lessons learned from elsewhere within the organization.

Within the process, HL7 has primary responsibility for (1) identifying and prioritizing services as candidates for standardization; (2) specifying the functional requirements and conformance criteria for these services in the form of Service Functional Model (SFM) specifications such as this document; and (3) adopting these SFMs as balloted HL7 standards. These activities are coordinated by the HL7 Services Oriented Architecture SIG in collaboration with other HL7 committees, which currently include the Vocabulary TC and the Clinical Decision Support TC.

Based on the HL7 SFMs, OMG will develop “Requests for Proposals” (RFPs) that are the basis of the OMG standardization process. This process allows vendors and other submitters to propose solutions that satisfy the mandatory and optional requirements expressed in the RFP while leaving design flexibility to the submitters and implementation flexibility to the users of the standard. The result of this collaboration is an RFP Submission, which will be referred to in the HSSP process as a Service Technical Model (STM). HL7 members, content, and concerns are integral to this process, and will explicitly included in the RFP creation and evaluation process.

It is important to note that the HL7 SFMs specify the *functional* requirements of a service, the OMG RFPs specify the *technical* requirements of a service, and the STM represents the resulting technical model, except as specified below. In many cases, SFMs describe an overall coherent set of functional capabilities and / or define a minimum set of behaviors necessary to guarantee a minimal level of service in a deployment scenario. These capabilities may be specialized or subdivided from both functional and informational (semantic) perspectives to provide conformance “profiles” that may be used as the basis for the OMG RFP process and/or implemented.

1.1.2 Service Definition Principles

The high level principles regarding service definition that have been adopted by the Services Specification Project are as follows:

- Service Specifications shall be well defined and clearly scoped and with well understood requirements and responsibilities.
- Services should have a unity of purpose (e.g., fulfilling one domain or area) but services themselves may be composable.
- Services will be specified sufficiently to address functional, semantic, and structural interoperability.
- It must be possible to replace a conformant service implementation with another one meeting the very same conformance profile while maintaining functionality of the system.

A Service at the SFM level is regarded as a system component; the meaning of the term “(system) component” in this context is consistent with UML usage[1].A component is a modular unit with well-defined interfaces that is replaceable within its environment. A component can always be considered an autonomous unit within a system or subsystem. It has one or more provided and/or required interfaces, and its internals are hidden and inaccessible other than as provided by its interfaces.

Each Service’s Functional Model defines the interfaces that the service exposes to its environment, and the service’s dependencies on services provided by other components in its environment. Dependencies in the Functional Model relate to services that have or may in future have a Functional Model at a similar level; detail dependencies on low-level utility services should not be included, as that level of design is not in scope for the Functional Model.

The manner in which services and interfaces are deployed, discovered, and described is outside the scope of the Functional Model, and must explicitly be dealt with in implementation. However, HSSP Functional Models may reference content from other

[1] It is expected that services will be defined, in response to the OMG RFP process, as UML components, however that level of design is outside the scope of the Functional Model.

areas of HSSP work that deals with architecture, deployment, naming and so forth. Except where explicitly specified, these references should be considered informative only. All other interactions within the scope of the scenarios identified above are in the scope of the Functional Model.

Reference may be made to other specifications for interface descriptions, for example where an interface is governed by an existing standard.

1.2 Overall disclaimers

- Examples and diagrams are illustrative and not normative unless otherwise specified
- The scope of information content of HSSP service specifications is not limited to HL7 content models. At a minimum, however, instances of an RLUS service should provide a semantic profile as part of its conformance profile to provide support for HL7 content models where applicable.

1.3 Context of this SFM within HSSP Process

The RLUS specification is a product of HSSP's Service Specification Development Framework (SSDF), and as such, has been vetted through the project's members as representative of the interests of the healthcare community. It is a primary component in the evolving HSSP infrastructure for healthcare, and defines primary interoperability functionality.

2 Service Overview and Business case

2.1 Service Overview

2.1.1 Service Description and Purpose

The Retrieve, Location, and Updating Service (RLUS) provides a set of interfaces through which information systems can access and manage information. RLUS allows health data to be located, accessed and updated regardless of underlying data structures, security concerns, or delivery mechanisms.

RLUS explicitly occupies the service space within an information processing environment. It is independent of but compatible with underlying structures, including local security implementations, data models, or delivery mechanisms. By separating and exposing those aspects of resources that facilitate inter-organization work flows in a service layer, this specification abstracts the problem of interoperability away from underlying systems. It is this abstraction and reconfiguration that allows interoperability and system durability independent of burdensome technology integration.

2.1.2 Scope

The Retrieve, Location, and Updating Service (RLUS) functional model specification seeks to define, at a service level, appropriate interfaces to locate, retrieve, and update resources among and between healthcare organizations. It is not intended to replace existing systems or implementations, but to create an interface standard for a service-oriented layer to expose those healthcare assets and resources within an organization that are needed to meet business or medical needs.

The manner in which services and interfaces are deployed, discovered, etc. is outside the scope of this functional model. However, it is expected that HSSP conformant services and specifications will utilize a number of implementation guides that will be made available over time. All other interactions within the scenarios identified in Section 2 are considered in scope.

2.2 The reason why the service is necessary

2.2.1 Rationale

Services in general make individual systems and components less brittle and more durable (e.g., less subject to change) because they guarantee a minimum level of functionality internally and externally to an organization. By providing for functionality inherent in a location, retrieval and update service, an essential component of interoperability is thus defined. Further, by explicitly creating this component as a service, the need to adhere to the system architecture being used by the organization is reduced to aspects implied in the service interface definition, allowing narrowly defined roles and functionality to be supported between organizations.

RLUS supports extended business cases that are common to healthcare, but which have either gained little support in local implementations or are implemented in ad hoc ways. These business cases generally involve multiple partners in the health team sharing data over long periods of time. RLUS provides an appropriate space to define and implement these business cases, because the overall effects on internal systems are minimized.

This specification approaches this wide scope by creating several measurable and testable levels of interoperability (see Section 6). These conformance levels allow for an RLUS compliant service to respond to different business partners according to different security models and partitions. This flexibility in turn provides the basis for Service Level Agreements (SLAs) between sharing partners and organizations. Currently, SLAs are written in an ad-hoc manner and are derived from an arbitrary starting point. The RLUS model provides a clear starting point on which these agreements may be based.

2.2.2 Vendor Viewpoint and Potential Business Opportunity

Within a mobile society, healthcare information systems' value will ultimately be derived from their ability to locate and supply healthcare resources found in other systems and different formats. RLUS takes the volatile healthcare market place as a given, and provides a primary interaction mechanism between systems, remaining independent but complementary to underlying technologies.

RLUS gives vendors several opportunities:

1. RLUS interfaces could be a fundamental component of any distributed health record system.
2. RLUS interfaces may be included as a functional set of service interfaces for their healthcare applications.

3. Vendors can add value to the customer’s solution with the addition of one or more RLUS-conformant services. This increases the scope of each customer’s reach, and in turn allows additional value to be available to other clinical practices

2.2.3 Target Audience and Added Value

RLUS provides architects and designers with an effective mechanism to specify, scope, and implement the portions of a healthcare enterprise architecture committed to interoperability. Whether this interoperability is between departments, systems, or organizations, RLUS embodies a flexible, responsive methodology that shortens design time and reduces product iterations.

An RLUS interface occupies the portion of the technology stack that is dedicated to the service space. As a service or a set of services, RLUS provides for low-cost, easily maintained components that can be molded to fit most deployment scenarios. Without specifying particular delivery mechanisms, payloads, or technologies, a clear strategy is nevertheless outlined that lays the groundwork for bridging the gap between information systems.

One of the chief methods to this end is the RLUS conformance profiles. Conformance profiles allow for an organization to align with different trading partners according to different trading rules, as realized by functional and semantic profiles. Simple interfaces may conform to different deployment scenarios while still remaining common in their pedigree.

Conformance to the RLUS Functional Model, then, provide the fundamentals for implementing an organization’s business strategies that extend beyond core business components. Interoperability services become an effective means by an organization to meet the shifting industry or trading requirements that characterize the healthcare industry. As such, RLUS-compliant services reduce implementation costs for an organization while serving to meet the current and evolving interoperability strategies imposed both from above and below.

2.3 Structure of the Service

In order to provide for the maximum implementation flexibility, this functional model defines several enumerated functional profiles for RLUS. These profiles identify a subset of the RLUS available functionality as pertinent to a specific context.

- Administrative – The interfaces contained in this profile define the service groupings necessary for minimal maintenance functionality.
- Location – This profile allows consumers of the service to obtain the metadata pointing to resources within a steward organization.
- Retrieval - allows resources to be retrieved.
- Updating - allows underlying repositories to be managed and adjusted according to well-defined informational constraints

- Locate and Retrieve (Extended) – allow functionally and semantically rich interactions, including nested queries, query by alternate semantic signifier, and describe nested semantic structures within resources. For example, could be used in conjunction with decision support services.

These profiles are detailed in Section 6.

The degree to which an organization's interoperability deployment supports a conformance profile, then, is directly related to the other agreements implemented with a business partner. A single RLUS service may respond to different real-world business partners depending on the underlying agreements and needs. For example, an organization may implement an RLUS (Updating) compliant service with a trusted partner. A separate trading partner may only be allowed RLUS (Location) access as dictated by other factors. For a discussion on how services co-exist in a various deployment scenarios, please see Section 2.4.1.

Additionally, RLUS explicitly makes no distinctions at the functional level regarding semantics of the underlying systems. Instead, it provides for a semantic profile as part of RLUS conformance profiles. This allows definition, publication, and discovery of vital semantic artifacts between sharing partners through RLUS interfaces without requiring strict, tightly coupled integration. Thus, RLUS does not preclude a strategy for semantic interoperability to be realized, though it would likely depend on other factors (for example, a common terminology service and / or a transformation service). This improves RLUS as an interoperability mechanism by relegating the issue of semantic interoperability to the trading partners, allowing semantic transformations to be performed at the least cost for the most derived value.

RLUS may serve a mission outside of the scope of the defined conformance profiles. To that end, several business cases, interfaces, and metadata elements have been defined that suggest an RLUS service that may be compliant to a profile, and still serve as a valuable service to other healthcare mechanisms, such as automated decision support.

2.4 Implementation Considerations

2.4.1 Deployment Scenarios

From the standpoint of this standard, any RLUS deployment must meet these criteria:

- It must simplify interoperability between organizations, systems, or departments
- It must allow conformance to this standard to be testable and verifiable
- It should be semantically extensible to meet expanding business needs

RLUS is explicitly an interface specification, not an implementation specification. As such, it is intended to be an interoperability mechanism between organizations. There is nothing inherent in the specification that precludes its use within a single organization, allowing a standardized method of record registry, location, and access. Conformance to the specification is asserted against profiles (see below) of the specification rather than against the specification itself.

Thus, locally, there is nothing to preclude RLUS being used to expose one or many internal registries or repositories. It can work in multiple different deployment topologies, and can be used to support different types of information. These are all deployment decisions and deployment context sensitive, and are valid insofar as they are explicitly profiled.

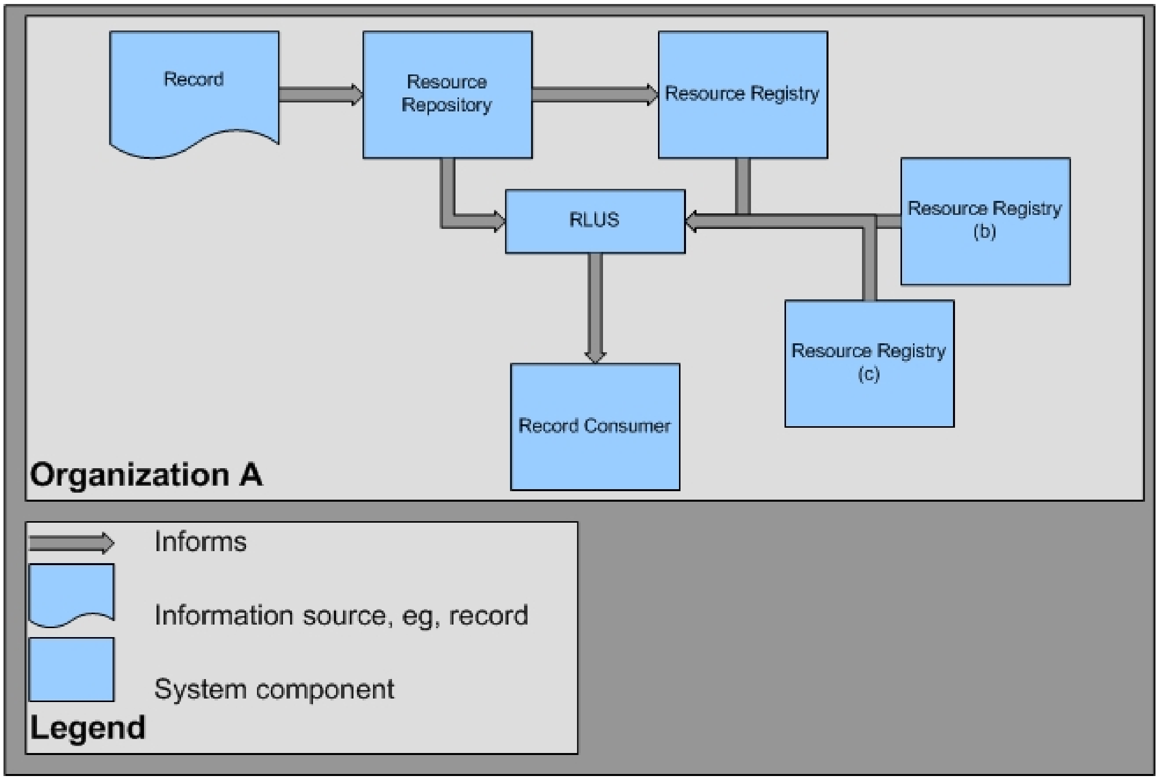


Figure 1: Representative deployment of RLUS within an organization (intra-organizational). Information is passed from source to system components, and then to consumers.

Figure 1 describes an intra-organizational deployment scenario where RLUS takes the role of enterprise registry, wherein the location of data is stored. Data, in turn is stored in repositories. However, RLUS may also be used in an inter-organizational setting where it is provisioned by more than one organization's systems, such as security or entity identification (See Section 9 for a more thorough discussion of cross-organizational RLUS). This could be a regional or a national metadata registry, or it could be a service set up between 2 or more data sharing partners. It could, in other words, function as an intermediary between organizations or systems.

In this setting, there is no additional functionality that must be mandated for RLUS to fulfill its mission. Deployment becomes more a matter of policy, agreement, and ownership than of technical consideration. This is especially true if RLUS is to handle repository updates in one or more systems. For a more thorough discussion of the related issues of RLUS federation and Service Discovery, see Section 9.7.

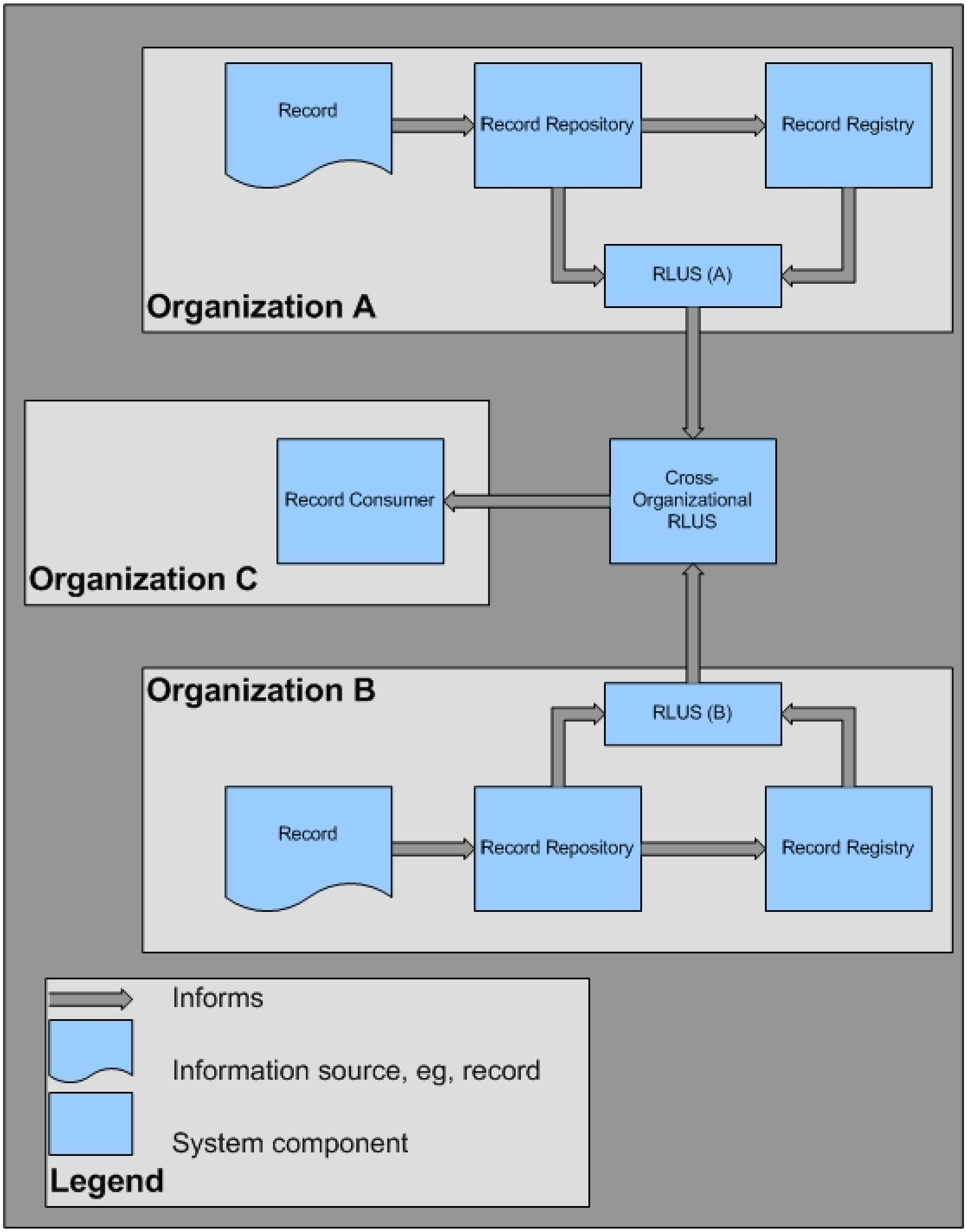


Figure 2: Representative RLUS deployment in a cross-organizational setting (inter-organizational). Cross-organizational RLUS implementations are discussed in the glossary.

RLUS excludes from its scope some interoperability components whose functions are assumed to either exist or be provided elsewhere, including security, terminology resolution, identity management, underlying data structures, or specific implementations of record storage, registry, or creation. [Note: For the purposes of this specification, those components that may be provided by the HSSP project are beyond the scope of the specification as they are part of the deployment environment.] While RLUS maintains no formal dependencies on these areas, RLUS implementations may elect to take advantage of their existence when those functions are available. For example, cross-organizational RLUS would depend on a cross domain entity identification service, which might conform to the emerging HSSP's XEIS specification.

Interactions between the service provider and service consumer must remain consistent across deployments, allowing functional interoperability according to explicit conformance profiles. See Section 2.4.3 for a discussion of conformance.

The semantic signifiers, functional profile, and supporting business cases are constrained to define the minimum implementation that may be expected to be implemented and at the same time be conformant. Extended interoperability is defined by extending these three elements to allow richer interactions. Semantic Signifiers are particularly extensible. New ones may be added to an existing deployment because RLUS' functional interfaces are not tightly coupled to informational semantics.

2.4.2 Semantic Signifiers in a Deployment Context

Semantic Signifiers provide for facilitating a meaningful interchange of information between transactions involving RLUS. RLUS stops short, however, of mandating the specific information content to be carried by the service for several reasons. First, this provides the implementer the ability to use the information semantics that are most appropriate for their needs. While infrastructure is being put into place for industry consensus-building around standard semantic signifiers (and implementers are encouraged to use them), this is not mandated.

Semantic Signifiers within the base specification are expressed as a data type allowing for platform-level binding of RLUS while keeping open a construct allowing for scalability, extensibility, and diversity of semantic information. Additionally, RLUS subscribes to the HSSP's profiling mechanism to allow for strong conformance assertions to be made, inclusive of informational semantics and designated semantic signifiers. This approach allows RLUS to carry payloads that have been standardized by other specifications or groups (e.g., HL7 v3). The RLUS specification includes only a minimal semantic profile to insure its ability to be effective within a deployment scenario. See Section 6 and Section 13 for details.

At run-time, a semantic signifier is the mechanism for realizing semantic profiles. This element may be an HL7 artifact, a locally published template, a nationally published template, a published XML schema, or to an agreed upon set of values. It may be passed by reference or by value, as both satisfy the functional requirements and meet the business needs. However, it is vital that a deployed RLUS can describe information about the semantic profile or profiles that it supports. Depending on the functional profile supported in a deployment, this could include:

- that it delineates specifically the semantic signifiers by which queries may be made
- that it delineates specifically the semantic signifiers by which responses will be delivered

- that it supports queries of one resource from another resource where both are described by semantic signifiers.

Semantic transformation or adaptation may happen within the requestor's domain, the responder's domain, or be included in the RLUS implementation, depending on agreements between trading partners and deployment context. See Sections 4.2.3, 6, and 9.1 for a discussion of semantic signifiers.

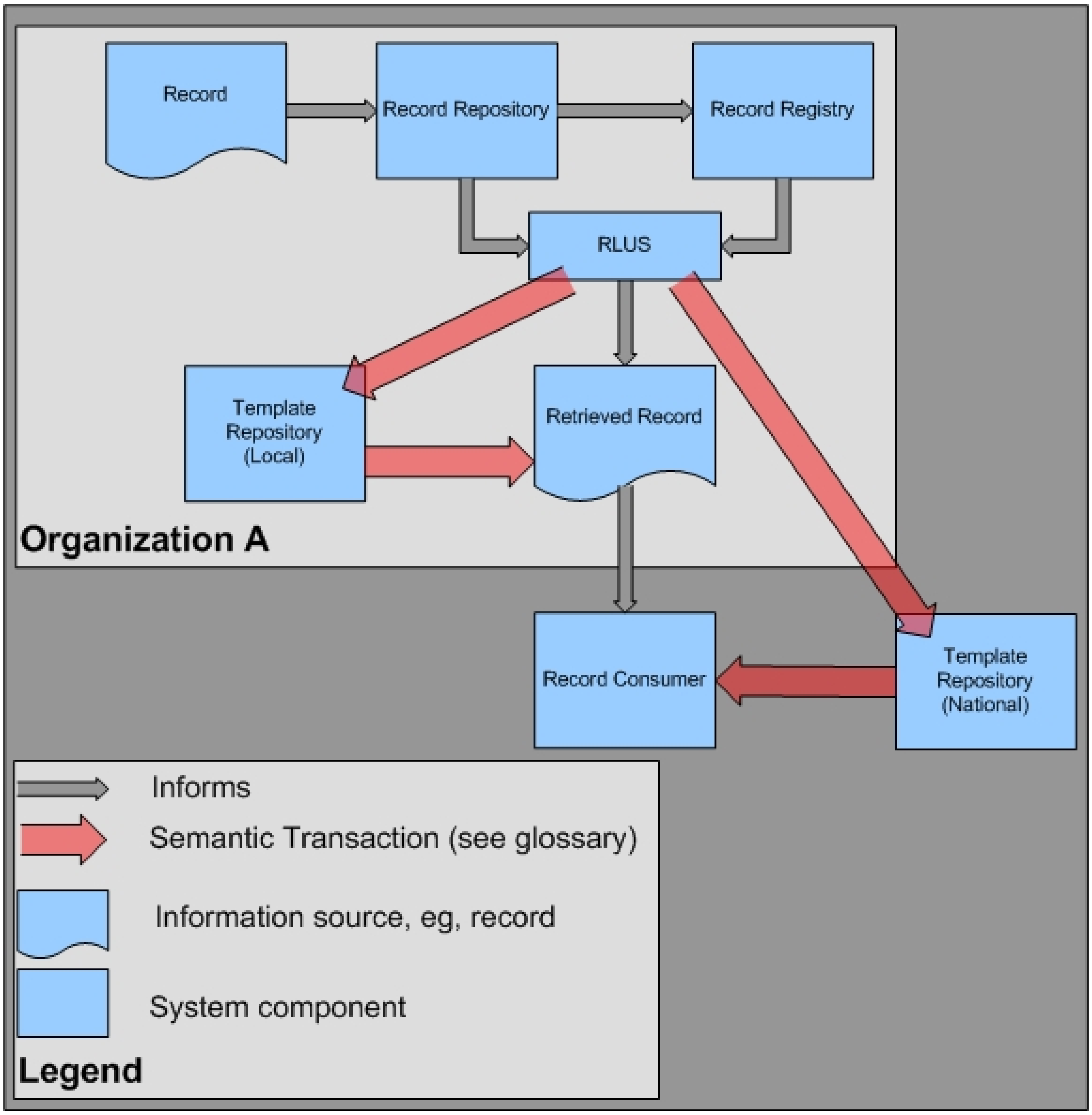


Figure 3: Representative RLUS Deployment as involved in a transaction with semantic exchange. Semantics are defined locally or globally, and are explicitly part of the business scenario.

2.4.3 RLUS Conformance

Conformance to this specification happens by way of conformance to profiles, which are detailed in Section 6. Conformance Profiles consist of at least one Functional (or Behavioral) Profile and at least one Semantic Profile.

Functional Profiles are the aggregation of all operations that are supported by a particular instance of RLUS. Likewise, Semantic Profiles are the aggregation of all Semantic Signifiers that collectively describe the semantics of the interactions supported by a particular RLUS instance.

3 Business Scenarios

3.1 Scenario Actors

The following actors take a role in the business scenarios:

- RLUS Consumer – The simplest illustration of an RLUS beneficiary is defined as a caregiver seeking information about a patient.

The following system components take a role in the business scenarios:

- Local Resource Service (RLUS) – The Retrieve, Location, and Updating Service acts as a mediator in location, retrieval, and updating transactions.
- Organizational Resource Repository (or repositories) – Contains assets, information, data, and other objects that are interesting to RLUS consumers. It is representative only of functionality without a normative restriction on the nature of information within an organization.
- User System – Any healthcare information system component that interacts with the user and which interfaces with other system components, including RLUS, to fulfill user needs. Includes record systems, inpatient systems, and other automated tools.

3.2 Primary Scenarios

All scenarios herein should be considered non-normative with regard to conformance to the RLUS standard. They are offered for explanatory purposes only. However, each primary scenario is tied to one or more interfaces, and as such, are suggestive insofar as a functional conformance profile will be met. Semantic profiles are considered interchangeable with regard to these business scenarios, and should be considered illustrative for purposes of demonstrating capability and scope.

Additionally, these scenarios are not intended to preclude deployed instances, localization, or extension. For example, some sort of tracing or auditing would likely be implemented to support versioning. As this concept speaks to a specific implementation condition, it is not included in the following scenarios.

3.2.1 Locate Resource

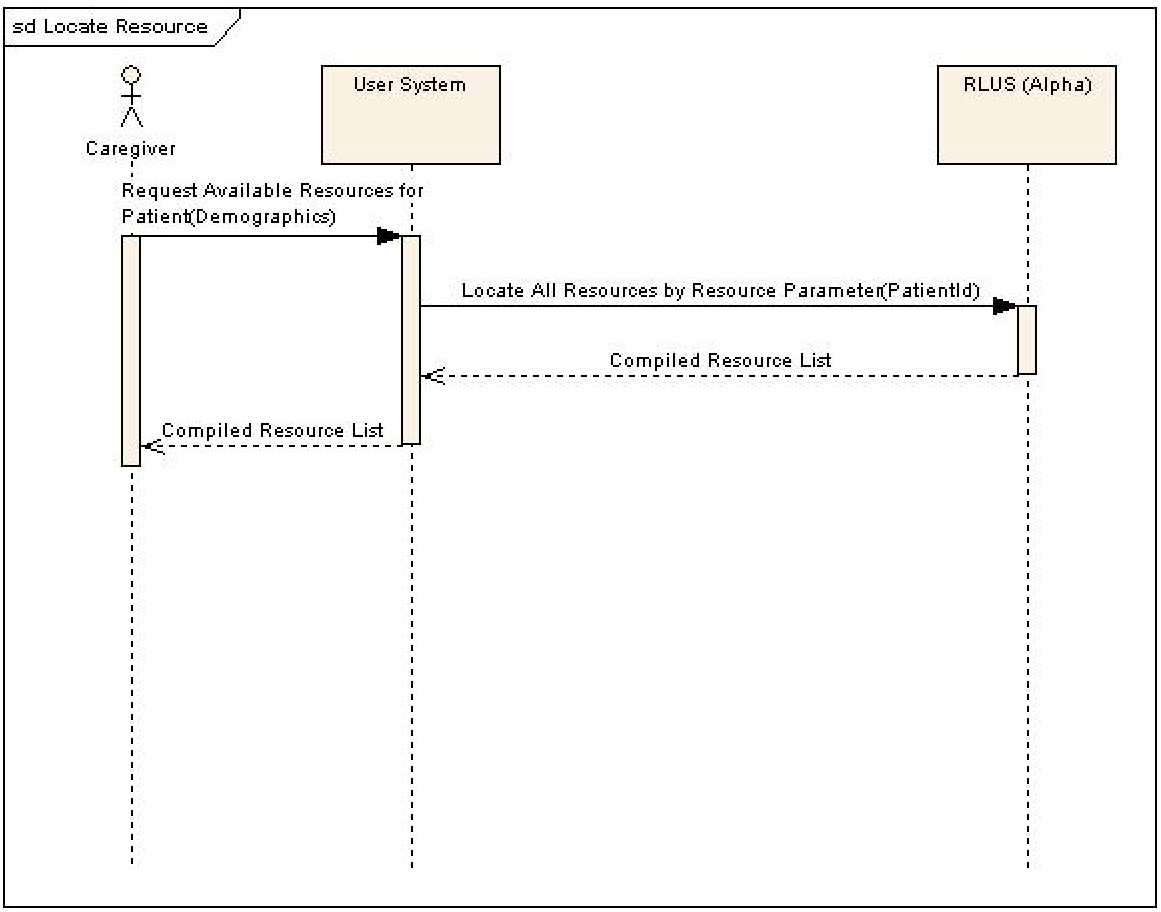
3.2.1.1 Scenario

A caregiver is seeing a patient whom she has seen many times. She accesses a list of all resources that are available for this patient within the local organization. The local RLUS implementation is informed by two separate local organizational registries. RLUS compiles a list of available resources and makes it available to the caregiver.

3.2.1.2 Interfaces Used

Locate Resources by Resource Parameter (Section 5.3.1)

3.2.1.3 Interaction Details



3.2.2 Retrieve Resource

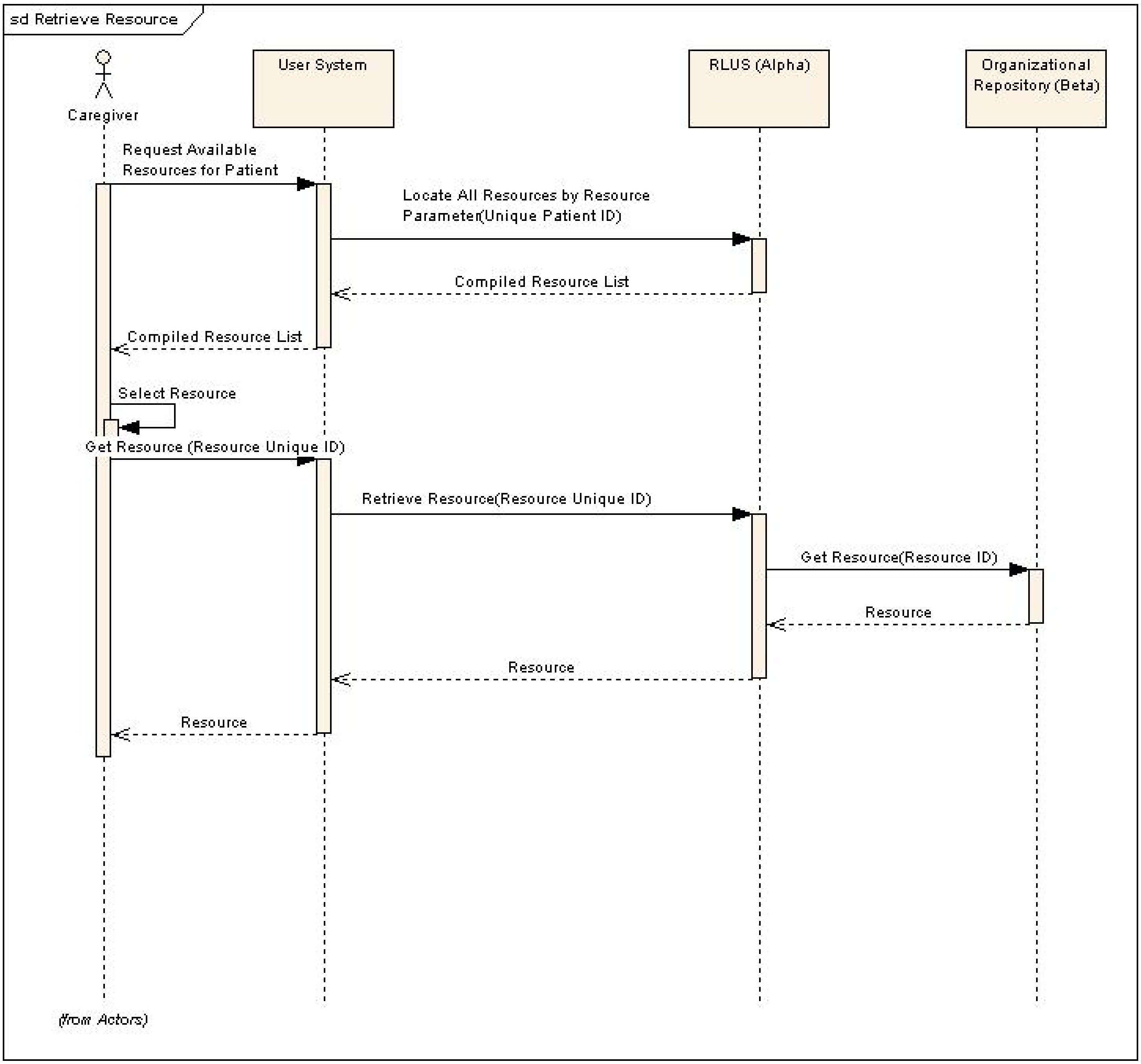
3.2.2.1 Scenario

A caregiver is seeing a patient whom she has seen many times. She accesses a list of all resources that are available for this patient within the local organization. The local RLUS implementation is informed by two separate local organizational registries. RLUS compiles a list of available resources and makes it available to the caregiver. Upon selection from a list, she retrieves the resource for viewing.

3.2.2.2 Interfaces Used

Retrieve Resource (Section 5.3.2)

3.2.2.3 Interaction Details



3.2.3 Retrieve Resource in Specified Format

3.2.3.1 Scenario

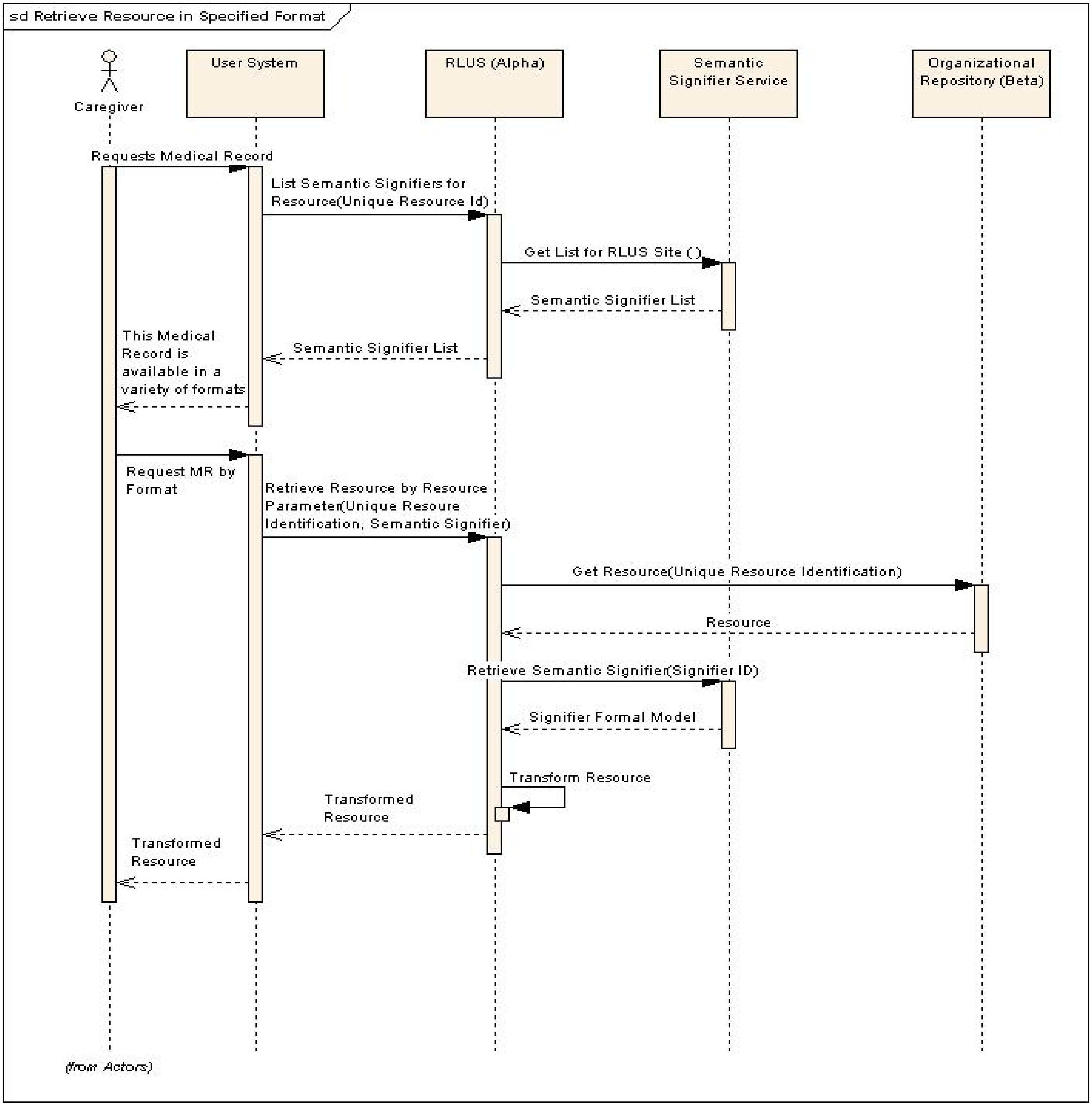
A caregiver is seeing a patient in an emergency room. Via RLUS, he accesses a list of all resources that are available for this patient from a separate organizational registry that stores metadata on the CDA-formatted documents. His local system first captures at run time the semantic signifiers that are available for this resource, and uses that list to create a user's list to be presented to the caregiver. As this is an emergency situation, the provider requests recent health record summaries rather than historical details, and indicates this preference when he makes the information request. His local system then compiles a list of available resources and makes it available to the caregiver. Upon selection from a list, he retrieves the patient's data that has been transformed into health record summaries.

3.2.3.2 Interfaces Used

List Semantic Signifiers for Resource (Section 5.2.2)

Retrieve Resource by Resource Parameter (Section 5.3.3)

3.2.3.3 Interaction Details



3.2.4 List and Retrieve Semantic Signifiers

3.2.4.1 Scenario

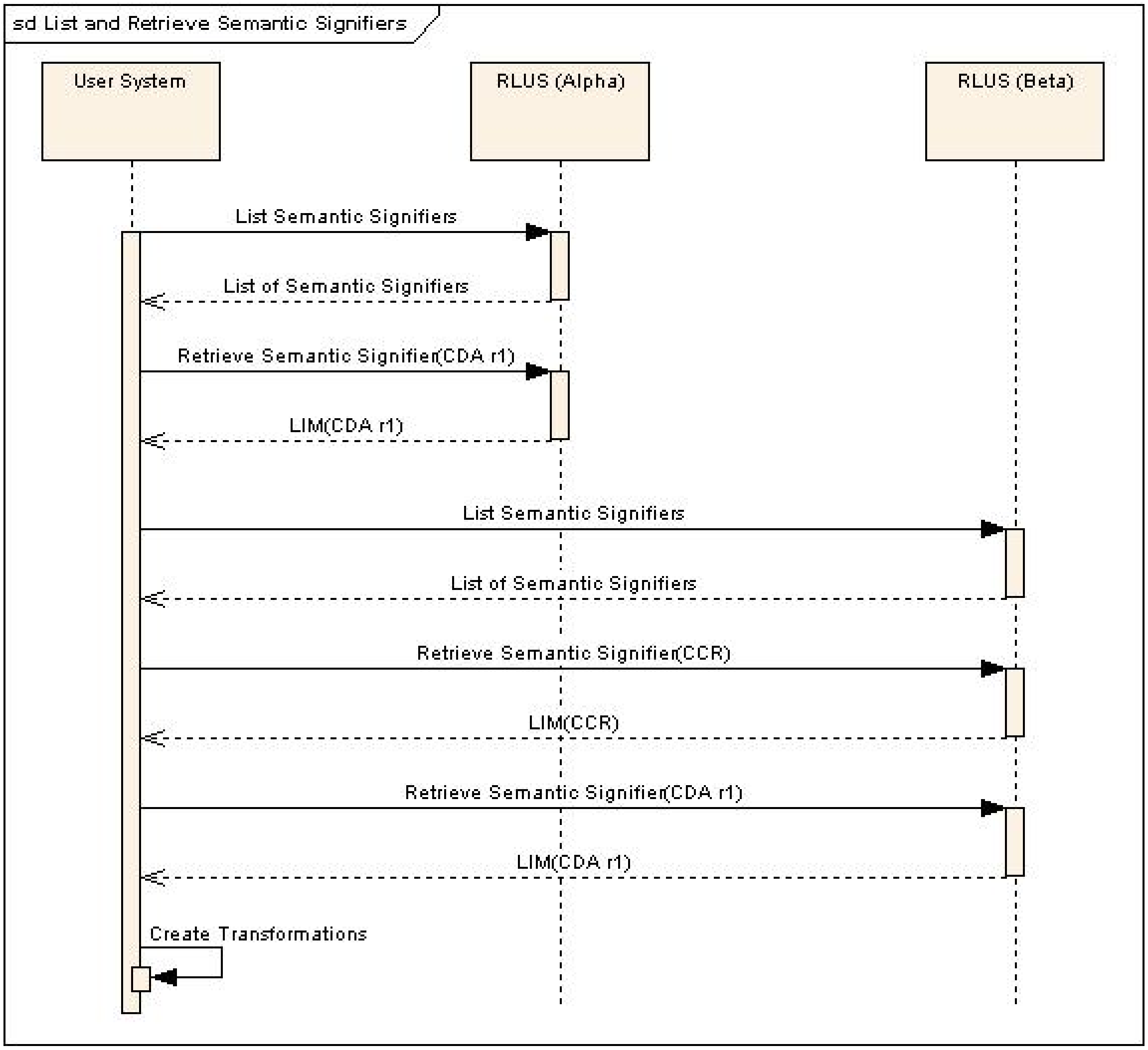
An advanced EHR system is tasked with integrating two organization's health information systems. As both systems expose RLUS (Retrieval) conformant service interfaces, the task is straightforward and algorithmic. The EHR system queries each service's self-description interface and discovers that Organization A supports CDA R1 and Organization B supports Continuity of Care Records (CCR) as well as CDA R1 for their retrieval interfaces. A further query to the systems actually retrieves the semantic signifiers themselves. The EHR is now aware of the semantics of the two systems, and may create an automated transformation between the two or notify a system analyst that further interaction is necessary.

3.2.4.2 Interfaces Used

List Semantic Signifiers (Section 5.2.1)

Retrieve Semantic Signifier (Section 5.2.3)

3.2.4.3 Interaction Details



3.2.5 Locate and Retrieve Information Using Resource Semantic Signifiers and Query Parameter Semantic Signifiers

3.2.5.1 Scenario

A clinician is seeing a patient with diabetes at an outpatient clinic. The clinician opens the “diabetes management” module in the EHR system. In populating this screen, the EHR system uses the local RLUS to retrieve the laboratory tests relevant to diabetes management, which include the patient’s hemoglobin A1C tests from the past two years, cholesterol panels from the past two years, lipid panels from the past two years, blood glucose tests from the past three months, and urine protein tests from the past two years.

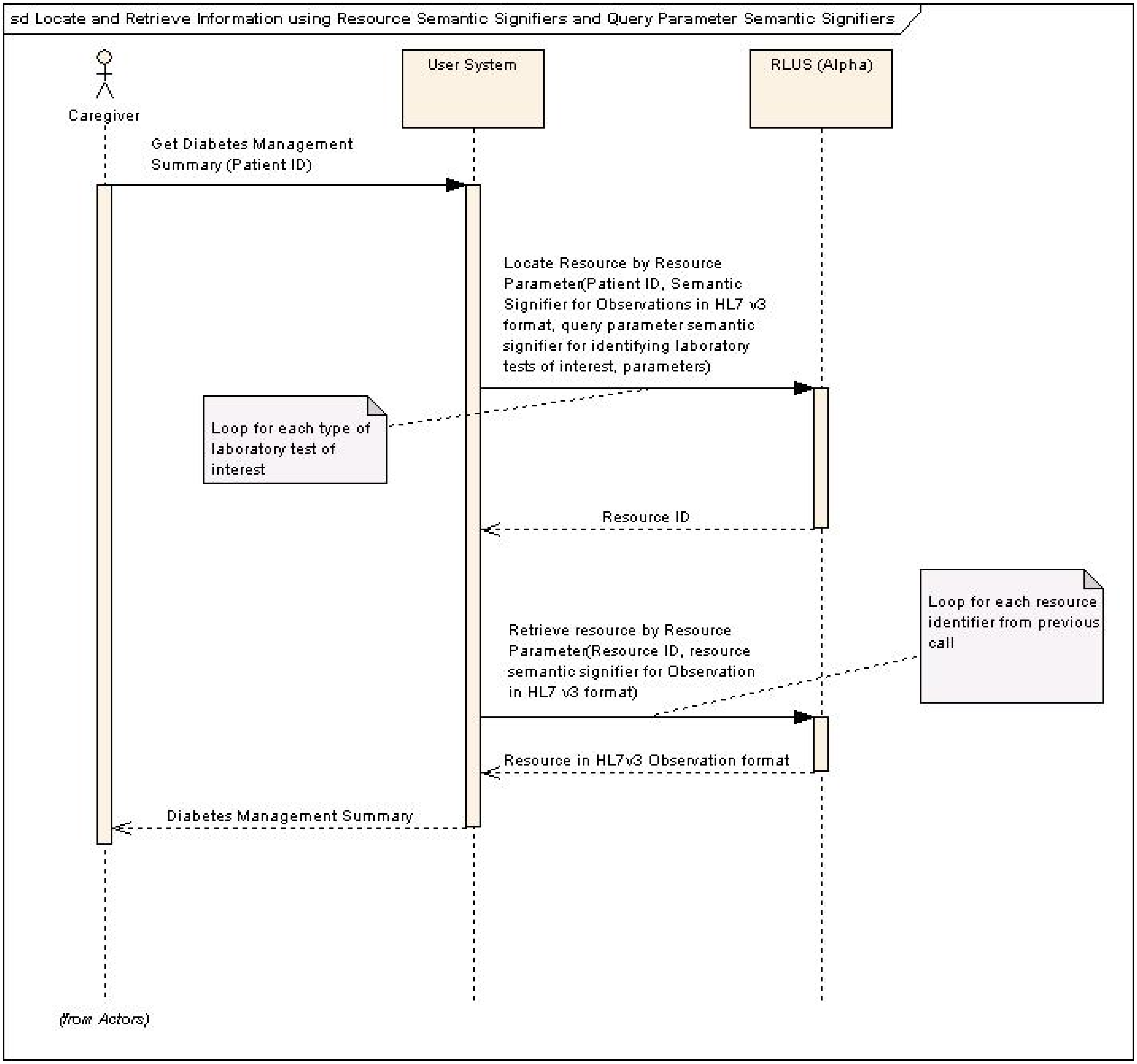
Note: the described interactions would likely have several dependencies, likely including a decision support system or some other aggregating element. This relationship is illustrative only, and is intended to clearly show that RLUS can encapsulate complex behavior that varies in scope and implementation.

3.2.5.2 Interfaces Used

Locate Resource by Resource Parameter (Section 5.3.1)

Retrieve Resources by Resource Parameter (Section 5.3.3)

3.2.5.3 Interaction Details



3.2.6 RLUS Entry Creation and Update

3.2.6.1 Scenario

A patient is discharged from an inpatient facility. Three discharge summaries are transported from the in-patient system to a record system in the hospital, which combines them into a chart to describe the patient’s stay. RLUS is notified that this chart is available instead of the individual discharge summaries. By agreement, this chart is the most granular element being shared with partners through RLUS.

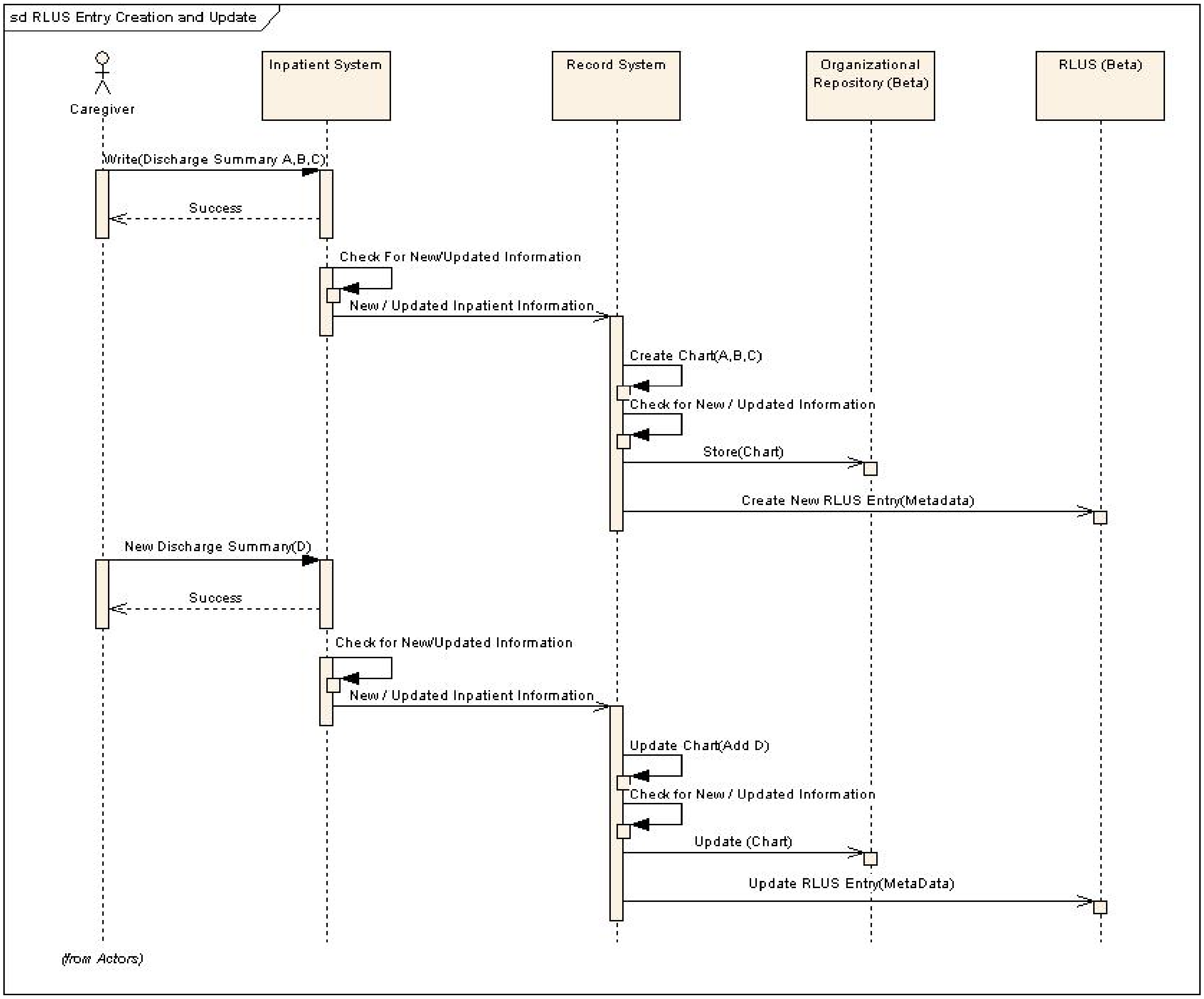
Three hours later, a fourth discharge summary is available for this patient, as it has retroactively added by one of the attending physicians. Once transported to the record system, a new version of the chart is made available that contains the four authoritative discharge summaries, replacing the older version. RLUS is updated to show a newer version of the chart is available with new information. The deprecated resource is given a forward reference to the newer chart, while the new chart is given a pointer to the deprecated resource.

3.2.6.2 Interfaces Used

Create an RLUS Entry (Section 5.1.1)

Update an RLUS Entry (Section 5.1.2)

3.2.6.3 Interaction Details



3.2.7 Resource Creation, Update, and Retrieval

3.2.7.1 Scenario

A patient is seen at a community clinic (Clinic A) with no information storage capabilities. The caregiver uses a web based user system to create medical records that are stored within Organization B's repositories, updating the RLUS (Beta) at the same time. The visit is demarcated within RLUS using the semantics that define the resource locally (e.g., start time, stop time, practice setting code, and healthcare facility).

A lab (Lab A) has a similar arrangement with Organization B. Using start time, stop time, and healthcare facility, a caregiver at Lab A pulls up the resources through RLUS, and adds the lab results to the document.

In the update above, the resource is returned to RLUS, where the metadata is updated, especially the status field that notes that the lab's revisions replace the initial notes. The older RLUS entry replaced, while the repository holds two revisions of the document: the initial plus the lab's additions. The older versions of the document contain forward references as necessary, while the newer version contains pointers to the deprecated resource. Thus RLUS acts as an intermediary between systems and storage, as well as handling resource creation, retrieval, and updating.

3.2.7.2 Interfaces Used

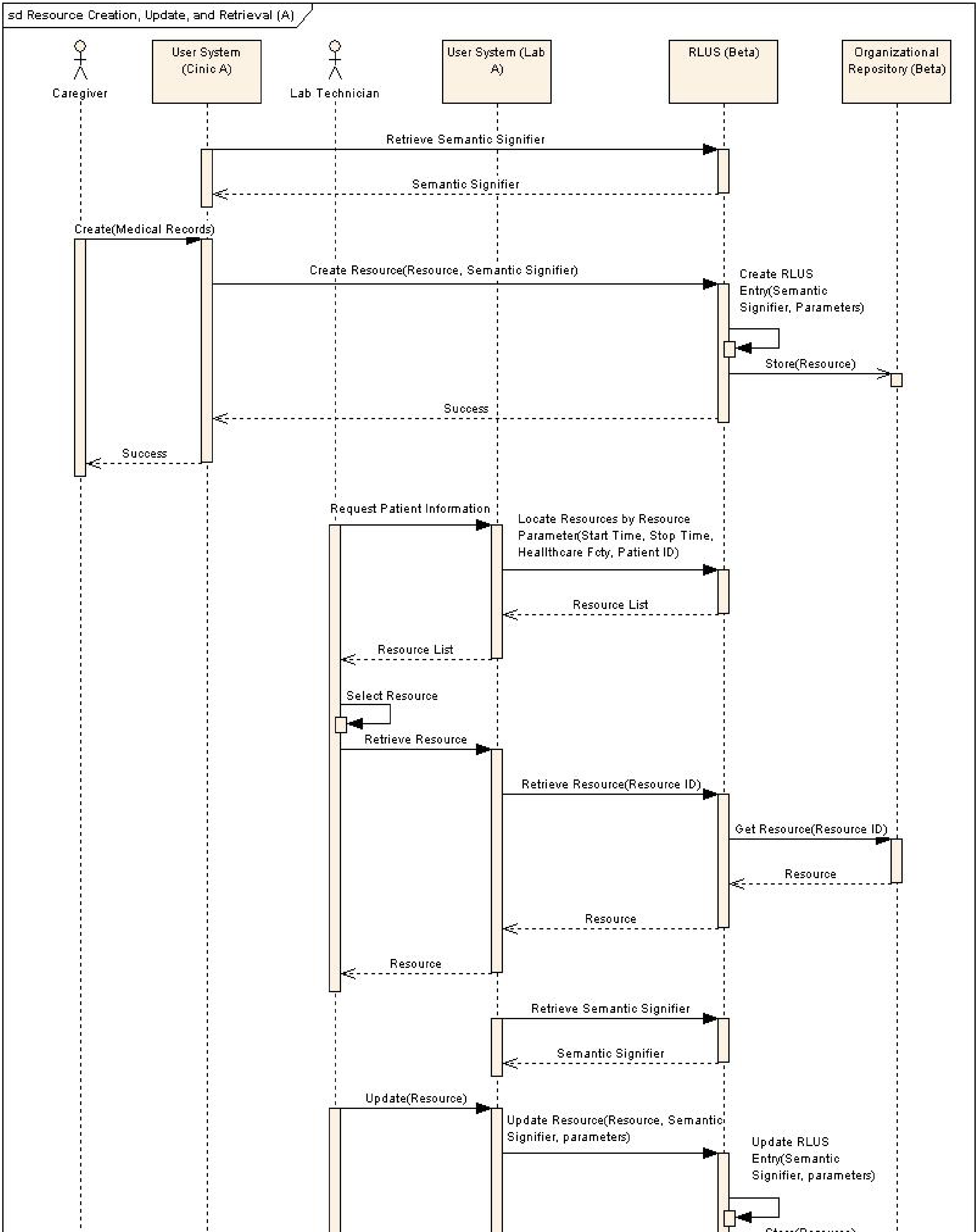
Retrieve Semantic Signifier (Section 5.2.3)

Create Resource (Section 5.3.4)

Update Resource (Section 5.3.5)

Retrieve Resource (Section 5.3.2)

3.2.7.3 Interaction Details


sd Resource Creation, Update, and Retrieval (A)
Caregiver
User System (Cinic A)
Lab Technician
User System (Lab A)
RLUS (Beta)
Organizational Repository (Beta)
Retrieve Semantic Signifier
Semantic Signifier
Create(Medical Records)
Create Resource(Resource, Semantic Signifier)
Create RLUS Entry(Semantic Signifier, Parameters)
Store(Resource)
Success
Success
Request Patient Information
Locate Resources by Resource Parameter(Start Time, Stop Time, Heallthcare Fcty, Patient ID)
Resource List
Resource List
Select Resource
Retrieve Resource
Retrieve Resource(Resource ID)
Get Resource(Resource ID)
Resource
Resource
Resource
Retrieve Semantic Signifier
Semantic Signifier
Update(Resource)
Update Resource(Resource, Semantic Signifier, parameters)
Update RLUS Entry(Semantic Signifier, parameters)

4 Assumptions and Dependencies

4.1 Assumptions

4.1.1 Resource Identification and Location

RLUS explicitly requires that each resource or asset that is going to be exposed via RLUS be uniquely and explicitly identified and located either explicitly or implicitly. If this is not explicitly part of the semantic signifiers used to represent the data, then these two informational components will need to be available to RLUS implicitly.

For example, RLUS could be used to expose Health Record Summaries available as discrete documents which have a physical location and an identifier. RLUS could also, though, be used to retrieve laboratory orders from a database, or from a composed set of services. In this case, the laboratory orders would not have a physical location per se, nor an identifier per se. Both instances of RLUS are valid depending on the semantic and functional profiles specified.

4.1.2 Entity Identification

Entities identified within semantic structures, such as patients, must have unique identifiers within a particular deployment context (for example, IHE's concept of *Affinity Domains*). For instance, Mrs. Jones is almost certainly uniquely identifiable in Organization A and in Organization B. Organization B needs to be able to uniquely identify Mrs. Jones as they access A's systems. This level of entity identification is supported by HSSP's Entity Identification Specification, though RLUS does not assume how this functionality is provided.

4.1.3 Information Quality

RLUS implicitly occupies a role of being the external gateway to authentic, authoritative data within an organization. However, RLUS can only guarantee the quality of the structure of the information rather than the information itself, for which it relies on the underlying systems. Put another way, RLUS rigorously adheres to functional and semantic profiles, but can only deliver information as well as the systems that it represents.

4.2 Dependencies

4.2.1 Dependencies on other HSSP Infrastructure Services

Although not formally specified, RLUS implementations can be greatly strengthened through the availability of strong supporting infrastructure capabilities. Standards such as the Common Terminology Service (CTS), the emerging CTS II specification, and work in progress such as the Entity Identification Service all form foundational elements that will benefit RLUS.

Despite these benefits, dependencies on external components are intentionally loose to provide for significant flexibility in use and in deployment. Organizations may or may not standardize on common infrastructure pieces. As a result, these dependencies are instead identified as pre-conditions throughout the specification, with the expectations that capabilities such as those being addressed in these foundational infrastructure services would address those needs. Organizations so choosing may omit these capabilities, or use internal or legacy alternatives so long as the specified preconditions are addressed somehow.

4.2.2 Cross-organizational RLUS and Discovery and Description Services

A cross-organizational implementation of RLUS is not another profile to be described. Rather, it is a special deployment case with unique parameters. There are no functional differences between RLUS implemented locally and a regional implementation that spans organizations. Each may operate at any one of the interface conformance levels, each may support the extended use cases, and so on. The difference is that a cross-organizational implementation has different qualitative dependencies.

For example, it will still need some sort of entity identification service, but it will need one that spans organizations. Similarly, its security profile will need to be able to interact with different systems in different way under different agreements.

See Section 10.7 for cross-organizational implementation details.

4.2.3 Support for Semantic Signifiers

RLUS requires the information that it exposes to conform to some structure that can be used for classification and organization. The term chosen for these structures is semantic signifier, and they are essential components of RLUS.

Semantic signifiers provide common information “building blocks” that RLUS interfaces use to convey content. Because RLUS exists at the service level, and because of the requirement for services to be both composable and self-contained, RLUS enables information that it exposes to be described in terms of these signifiers through the functional interfaces (e.g., there is a mechanism in the interface for the dynamic discovery of the information content and profiles supported by an RLUS implementation). RLUS supports rich semantics in both the request and response portions of its interface lifecycle.

For the purposes of RLUS, semantic signifiers have certain characteristics that allow simple interfaces to support rich semantic interactions:

- They should be expressible in a format so that consumers of an RLUS service can discover the notations required to access certain functional components (see Section 4.2.4)
- They should be describable by reference or by value through the RLUS interfaces so as to be verifiable
- Finally, these structures should be traceable to a particular semantic profile so as to be tested for conformance at either run time or design time.

For example, if patient data is intended to be searched by a patient identifier, then that identifier needs to be included in the semantic representation of that data so that it may be queried. While RLUS makes only minimal assertions regarding semantic signifiers, it is inferred that these same semantic signifiers will support the profiled functionality of RLUS.

4.2.4 Semantic Signifier Formalization

Where the content being exposed through RLUS is HL7 Domain content, RLUS normatively requires that a semantic signifier be formally described by a Localized Information Model (LIM). LIMs provide a consistent means of allowing complex informational semantics to be communicated through service interfaces using industry standards. They are useful because they can allow the service consumer to associate knowledge of the instance over and above the inherent semantics of the information in the instance. See Section 9.1.1 for a discussion of the use logical models and physical models in RFP submissions.

4.2.5 The Role of Semantic Signifiers in RLUS

Semantic Signifiers express information that is inherent in several portions of a typical RLUS interaction. They can serve as a return type in that they classify and describe the information that is returned through the RLUS interface. They may also be used to classify and describe input parameters for RLUS functionality.

Semantic Signifiers may or may not inherently describe the resources that are exposed through the RLUS interfaces, and in fact, there is no explicit need for these resources to be modeled by a formalism at all. Strictly speaking, all that is necessary is that a resource be expressible by a Semantic Signifier by way of some behavior.

5 Detailed Functional Model

Note that each “capability” may result in one or more individual operations in the technical specification resulting from an RFP submission.

Interfaces in this section are grouped according to broad capability classifications. These functions are grouped into functional profiles in Section 6.2.1 below that may not exactly match the capability classifications, but align with broad business needs for the purpose of profiles.

5.1 Administrative and Management Interface

Administrative and management interfaces are services that change the quantity or quality of meta-information being exposed by RLUS.

5.1.1 Create an RLUS Entry

Description	Creates an entry in RLUS that points to an asset or resource that is intended to be made available.
Precondition	• There is no entry in RLUS that pertains to a particular identifiable and accessible entity • There is a need to expose a particular resource through RLUS.

Inputs	• Resource Location • Resource Identifier • Semantic Signifier
Outputs	• Success / Failure notification
Post-conditions	A new set of metadata points to an underlying resource belonging to an identifiable entity
Exception Conditions	• The resource location is invalid, the identifier is not unique or is null, or the semantic signifier is not supported • Metadata already exists in RLUS • New RLUS entry does not point to a unique and identifiable resource
Aspects left to RFP Submitters	Manner of integration with identity management system, such as EIS The service administrator must interact with a security implementation How to check that an RLUS entry points to a unique and identifiable resource

5.1.2 Update an RLUS Entry

Description	Replaces an existing RLUS entry with an updated entry that points to an asset or resource that is intended to be made available.
Precondition	• The RLUS entry exists • The updated entry pertains to an identifiable and accessible entity • Some metadata has changed, forcing RLUS to be updated.
Inputs	• Resource Location • Resource Identifier • Semantic Signifier
Outputs	• Success / Failure notification
Post-conditions	• The updated set of metadata points to an underlying resource

	belonging to an identifiable entity • The metadata has changed from its initial configuration, yet remains valid • The deprecated entry is given a forward reference to the new entry, and the new entry is given a backwards reference to the replaced entry
Exception Conditions	• Inputs are invalid • RLUS entry does not exist • Updated RLUS entry does not point to a unique and identifiable resource
Aspects left to RFP Submitters	Manner of integration with identity management system, such as EIS The service administrator must interact with a security implementation How to check that an RLUS entry points to a unique and identifiable resource

5.1.3 Delete an RLUS Entry

Description	Removes an RLUS entry, or makes it inaccessible.
Precondition	• RLUS entry exists • There is no longer a need to expose this resource through RLUS
Inputs	• Resource Identifier
Outputs	• Success / Failure notification
Post-conditions	The metadata has been removed from RLUS, or is no longer accessible to queries.
Exception Conditions	• The RLUS entry that was deleted is still accessible to service consumers
Aspects left to RFP Submitters	• It may be appropriate to completely remove the metadata, or it may be desirable to simply make this metadata set inaccessible via a flag of some sort. • Auditing: an RFP Submitter will likely want to keep track of

	who is deleting what. • If RLUS contains historical information, then all RLUS entries will need to be removed / updated

5.1.4 List Conformance Profiles

Description	Produces a list of conformance profiles supported by this RLUS instance
Precondition	• Consumer has agreed to exchange data with the organization via RLUS • RLUS instance knows which conformance profiles that it supports (e.g., set by configuration)
Inputs	• None
Outputs	• List of conformance profiles, including identifier, version, registering authority, and description
Post-conditions	A list of the conformance profiles supported has been delivered to the requestor
Exception Conditions	• Conformance profiles are not available
Aspects left to RFP Submitters	• Means of defining and maintaining which profiles are supported by which RLUS Instance • Describing conformance assertions • Testing conformance assertions
Miscellaneous notes	As particular instances of RLUS will make assertions regarding conformance, it is desirable to allow RLUS to self-describe its conformance level at design time and run-time. This description in turn allows the conformance assertion to be tested. A code system will likely be needed, either globally or within each context / domain, which identifies conformance profiles

5.2 Semantic Interface

Semantic interfaces include service interfaces that delimit the underlying informational qualities of RLUS, including the semantics to use as query parameters, the semantics for return values, and the enumerated semantic structures available within a resource. These interfaces are vital to the implementation of RLUS as they are an essential aspect of RLUS' capability to self-describe. It should be noted, though, that their implementation may vary between design-time interfaces and run-time interfaces, depending on the deployment context.

5.2.1 List Semantic Signifiers

Description	Lists semantic signifiers that are available from this RLUS implementation. What is available, and how can a client retrieve it?
Precondition	• Consumer has agreed to exchange data with the organization via RLUS • Semantic signifiers are describable and available • Semantic signifier is able to be modeled logically
Inputs	• None
Outputs	• List of semantic signifiers that are available for the entire RLUS implementation, including ➢ Semantic signifiers that specify the format and semantics of resources that can be returned by the RLUS (resource semantic signifiers) ➢ For each of the resource semantic signifiers, zero or more semantic signifiers that specify the information models that can be used to query for the data of interest (query parameter semantic signifiers)
Post-conditions	A list of semantic signifiers and their descriptions has been delivered to the requestor
Exception Conditions	• List of semantic signifiers are not available
Aspects left to RFP Submitters	It is desirable to do more than simply list the signifiers. For example, each signifier may point to a schema, or to an explanation, or to an implementation page. In the case of structured documents, it may be necessary to point not only to the schema, but to the transformation that accompanies the schema.
Miscellaneous	Having a ready list and description of each semantic signifier used

notes	by an organization is a minimum requirement for all levels of RLUS conformance.

5.2.2 List Semantic Signifiers for Resource

Description	Lists semantic signifiers that are available for a resource within this RLUS implementation.
Precondition	• Consumer has agreed to exchange data with the organization via RLUS • Semantic signifiers are describable and available • Semantic signifier is able to be modeled logically
Inputs	• Resource Unique Identifier
Outputs	• List of semantic signifiers that are available for application to this resource, including ➢ Semantic signifiers that specify the format and semantics of this resource that can be returned by RLUS (resource semantic signifiers), including semantic signifiers that are contained within the resource ➢ For each of the resource semantic signifiers, zero or more semantic signifiers that specify the information models that can be used to query for the data of interest (query parameter semantic signifiers)
Post-conditions	A list of semantic signifiers and their descriptions has been delivered to the requestor
Exception Conditions	• List of semantic signifiers are not available • Semantic signifiers are not describable, or the descriptions are invalid
Aspects left to RFP Submitters	It is desirable to do more than simply list the signifiers. For example, each signifier may point to a schema, or to an explanation, or to an implementation page. In the case of structured documents, it may be necessary to point not only to the schema, but to the transformation that accompanies the schema.
Miscellaneous	Having a ready list and description of each semantic signifier used by an organization is a minimum requirement for all levels of RLUS

notes	conformance.

5.2.3 Retrieve Semantic Signifier

Description	Retrieves a description and formal model of a local semantic signifier.
Precondition	• Submitter has agreed to exchange data via RLUS with the organization • Semantic signifiers are describable and available • Semantic signifier is able to be modeled logically
Inputs	• Unique Semantic Signifier Identification
Outputs	• The description of the semantic signifier • The formal model expressing the semantic signifier as expressed in a LIM. • The Status of the semantic signifier
Post-conditions	The model of the semantic signifier is available for further operations
Exception Conditions	• semantic signifiers are unavailable • Formal Model is missing or unreachable • Semantic signifier has been marked as invalid
Aspects left to RFP Submitters	See Section 4.2.3 and 4.2.4 for a discussion of semantic signifiers. See Section 9.1 for a discussion of implementing semantic signifiers in a response to the RFP.
Miscellaneous notes	This is an aspect of the functional model that is a key to interoperability. There needs to be an automated way of accessing the formal models that describe an organization’s content so that information can flow.

5.3 Run-Time Interface

Run-time interfaces expose functional capability that is intended to be implemented to expose underlying functionality.

5.3.1 Locate Resources by Resource Parameter

Description	Given a particular parameter, returns a list of metadata of all resources and their locations that are available to the requestor within the registry.
Precondition	• The requestor has access to an appropriate query structure before making the request
Inputs	• Query by example Or • Semantic signifier identifying the information model that must be available for returning the resource of interest (resource semantic signifier) • Semantic signifier identifying the information model that is used to express the query parameters (query parameter semantic signifier) • Query parameters, expressed in conformance with the query parameter semantic signifier
Outputs	• List of metadata sets that point to unique resources parameterized by the inputs
Post-conditions	A list of resources parameterized by the inputs is available to the requestor
Exception Conditions	• The resource is identified ambiguously
Aspects left to RFP Submitters	Querying by Example would be filling out portions of the syntax for a particular semantic structure that would be applicable for a search. For example, taking a blank CDA document and filling in the Patient Identifier and submitting that in order to request similar CDA's from RLUS is appropriate. Querying by a separate semantic signifier involves using different query semantics than response semantics. For example, a CDA with a date range and patient Identifier might be submitted with the intent of retrieving a list of health record summaries. This service may be extended to include date ranges, for example, or to any other semantically significant search criteria. This service interface may also contain some defaults, such as defaulting to "status=NEW" for example.

	Patient Consent across organizational boundaries
Miscellaneous notes	The intent of this interface is to return a snapshot of resources that match a set of search criteria. There is no implied interaction with the resource (locking, etc.)

5.3.2 Retrieve Resource

Description	Given a unique identifier, returns a particular resource.
Precondition	• Resource Exists • Resource is available, is not locked, is not corrupt
Inputs	• Unique Resource Identifier
Outputs	• The resource
Post-conditions	A read only representation of the resource is available to the requestor's system
Exception Conditions	• The Resource is unavailable • The Resource ID is unable to be resolved
Aspects left to RFP Submitters	• By value or by reference • Patient Consent across organizational boundaries
Miscellaneous notes	This interface explicitly exposes a resource as read-only, returning the resource by value to the requestor. Other models of interaction, say that involve locking, are certainly possible, but extend RLUS in ways that are not necessary for interoperability, though they may be necessary for deployment scenarios.

5.3.3 Retrieve Resources by Resource Parameter

Description	Given a unique identifier and a semantic signifier, returns a particular resource transformed to the semantic structure in the semantic signifier.
Precondition	• Resources are available and accessible to RLUS • Submitter has agreed to exchange data via RLUS with the organization

	• Semantic signifiers are being used • Semantic signifiers are describable and available • Semantic signifier is able to be modeled logically
Inputs	• Query by example (for example, patient identifier) Or • Semantic signifier identifying the information model that for returning the resource of interest (resource semantic signifier) • Semantic signifier identifying the information model that is used to express the query parameters (query parameter semantic signifier) • Query parameters, expressed in conformance with the query parameter semantic signifier
Outputs	• The resource, transformed into the semantic model identified as an input parameter
Post-conditions	• A read only representation of the resource is available to the requestor's system • The resource has been transformed using the semantic signifier
Exception Conditions	• The Resource is unavailable • The Resource ID is unable to be resolved • The semantic transformation is invalid • The semantic signifier is invalid or unavailable
Aspects left to RFP Submitters	• By value or by reference • The actual method of transmission. Actually using a service interface to support content dissemination may not be the best use of services. • Patient Consent across organizational boundaries
Miscellaneous notes	This interface explicitly exposes a resource as read-only, returning the resource by value to the requestor. Other models of interaction, say that involve locking, are certainly possible, but extend RLUS in ways that are not necessary for interoperability, though they may be necessary for systems.

5.3.4 Create Resource

Description	Creates a resource in a resource repository, and creates an entry in RLUS to point to that resource.
Precondition	• Resource does not exist in the Repository • Repository's registry is available • There is a need to expose a particular resource through RLUS. • Semantic Signifier describing the resource is compatible with the RLUS Instance
Inputs	• Resource
Outputs	• Success / Failure Notification
Post-conditions	• Resource is located in the repository • Resource is available in the repository • RLUS Entry has been created • If appropriate, the repository's registry is updated
Exception Conditions	• Repository is inaccessible • Resource does not have a unique identifier and a unique location • The semantic structure of the resource is not supported by this implementation of RLUS
Aspects left to RFP Submitters	• Security and semantics are two areas that will require attention when an external interface deals directly with an organization's internal systems. • Transactional integrity between updating RLUS and updating the repository
Miscellaneous Notes	For a discussion of resources, please see the glossary. The nature of the resource is implementation sensitive.

5.3.5 Update resource

Description	Edits an existing resource that is available and if necessary the RLUS entry pointing to that resource.

Precondition	• The resource exists in the repository • The resource is available • The resource has changed, and both RLUS and the contents of an underlying registry require updating
Inputs	• Resource • Resource Unique Identifier
Outputs	• Success / Failure Notification
Invariants	• The submitter has all required permissions and access rights • RLUS is operational
Post-conditions	• Resource is located in the repository • Resource is available in the repository • RLUS Entry has been updated • The RLUS is updated with a backwards reference to the deprecated resource, and the deprecated RLUS entry is given a forward reference to its new version.
Exception Conditions	• The resource is locked, in use, or does not exist • The repository is unavailable • The updated resource does not conform to the semantic structures supported by this implementation of RLUS
Aspects left to RFP Submitters	• Updating a resource will need to implement some sort of versioning at the repository level • The decision will have to be made what data is stored in RLUS, whether it is all authoritative, or historical, with one metadata set explicitly replacing another. • Tracing or auditing • Transactional integrity between updating RLUS and updating the repository • It is considered best practice for an update to be implemented as a “replace” – this is reflected in the post-conditions.

5.3.6 Delete Resource

Description	Removes a resource from a repository, or makes it unavailable. Also removes the RLUS reference to that resource, or makes it unavailable.
Precondition	• The resource exists • The resource is available • There is no longer a need to make this resource available via RLUS • There is no longer a need for the resource
Inputs	• Unique Resource ID
Outputs	• Success / Failure notification
Post-conditions	• The resource has been removed from a particular repository, or it has been made unavailable, and is no longer accessible.
Exception Conditions	• The resource is unavailable, is in use, or is locked
Aspects left to RFP Submitters	• Whether the resource is actually removed or is simply made unavailable (updates) • It is likely that some sort of trace will be in effect to see what is being removed, by whom, why, etc. • Transactional integrity between updating RLUS and updating the repository

6 Profiles

6.1 Introduction

A set of profiles may be defined that cover specific functions, semantic information and overall conformance. The SDF explains in detail the meaning of each of these types of profile. In brief, they are as follows:

- Functional Profile: a named list of a subset of the operations defined within this specification which must be supported in order to claim conformance to the profile.
- Semantic Profile: identification of a named set of information descriptions (semantic signifiers) that are supported by one or more operations. In the case of RLUS, semantic signifiers express query parameters, return parameters, and enumerate resources within resources that are described by semantic signifiers.
- Conformance Profile: this is a combination of a set of functional and semantic profiles taken together to give a complete coherent set of capabilities against which conformance can be claimed.

Due to the fact that RLUS represents generic functional capability, none of the specific functional profiles are mandatory. For example, an RLUS instance could support only the Retrieval profile (similar to part of IHE's XDS) or only the Location profile. Additionally, functional profiles are additive, and may be present in multiplicity. However, any instance of RLUS must support at least one of the functional profiles defined below.

Semantic operations are defined in specific profiles only insofar as the discovery and description of the semantics underlying an RLUS implementation enable the business scenario. By design, RLUS interfaces are loosely coupled with the informational qualities that they expose, though this support is enabled through the mandatory interfaces within profiles. Semantic profiles, like Functional Profiles, may be present in multiplicity.

No specific normative conformance profiles have been defined in this specification. Therefore, there are no normative functional or semantic profiles. The creation of conformance profiles is detailed in the HSSP's Service Specification Development Framework guide (for more information on this guide and related materials, see the Infrastructure sub-group at http://hssp.wikispaces.com). Profiles are not normative except within the context of a conformance profile in a deployment context. Additional profiles may be created reflecting local requirements or on a larger scale.

6.2 Functional Profiles

The profiles below are not mandatory as a collection, but any implementation must support at least one of the profiles listed below. Note that these profiles cross the capability classifications in Section 5 to arrive at interface groupings that create business solutions. Profiles in general should provide for mixing and matching of capabilities to constrain this specification so that business needs are met and localization achieved.

Profile Name	**Included Functionality**
Location	5.1.4 List Conformance Profiles 5.2.1 List Semantic Signifiers 5.3.1 Locate Resources by Resource Parameter
Retrieval	5.1.4 List Conformance Profiles 5.2.1 List Semantic Signifiers 5.2.2 List Semantic Signifiers for Resource 5.2.3 Retrieve Semantic Signifier 5.3.2 Retrieve Resource 5.3.3 Retrieve Resources by Resource Parameter
Update	5.1.4 List Conformance Profiles 5.2.1 List Semantic Signifiers 5.3.4 Create Resource 5.3.5 Update Resource 5.3.6 Delete Resource
Administrative	5.2.1 List Semantic Signifiers 5.1.1 Create an RLUS Entry 5.1.2 Update an RLUS Entry 5.1.3 Delete an RLUS Entry 5.1.4 List Conformance Profiles
Locate and Retrieve (Extended)	5.1.4 List Conformance Profiles 5.2.1 List Semantic Signifiers 5.2.2 List Semantic Signifiers for Resource 5.3.1 Locate Resources by Resource Parameter 5.3.3 Retrieve Resources by Resource Parameter

6.3 Semantic Profiles

This version of the specification normatively defines only one semantic profile: Clinical Document Architecture, Release 1. This profile is considered optional since RLUS could expose varied other informational structures to address business needs, though the HL7 CDA semantics are applicable to all Functional Profiles in Section 6.2.

See Section 13 for a detailed description of the HL7 CDAr1 Document Profile.

6.4 Optional Functional and Semantic Profiles

The above profiles are not intended to meet every situation, business need, or implementation. They form the basis for establishing relationships between trading partners, and identify the bare minimum functionality that can help RLUS to meet its intended interoperability mission. Thus, the profiles are provided for organizations creating their own RLUS interfaces, for vendors including RLUS implementations in their products, and other standards that rely on RLUS to meet certain needs in fulfilling their own conformance criteria. Further, conformance to the minimal, non-normative functional and semantic profiles (HSSP RLUS HL7 CDA R1 Location, for example) is testable for compliance. This helps to insure a degree of interoperability and functionality across all implementations. See Section 9, and specifically 9.3, for a more thorough discussion of conformance, service longevity, and interoperability.

7 The Services Framework Functional Model

The Services Framework Functional Model identifies common underlying enterprise infrastructure such as naming, directory, security, etc. that may be assumed and referenced by this Functional Model.

Note that the Services Framework Functional Model is being developed in parallel with other service Functional Models; candidate functionality for the Framework should be submitted to the Infrastructure subgroup for evaluation.

RLUS-compliant service instances are intended to be healthcare middleware services and to work within the context of supporting infrastructure services that may exist within an enterprise. As a result, a number of underpinning capabilities have been intentionally omitted from the scope of this specification. These include (but are not limited to) capabilities such as identity management, security and terminology management.

The RLUS specification, by design, can be used as a means to integrate a new capability into a service-oriented architecture, or can be used to provide a service interface to access content in legacy applications. It is not intended as a replacement of any single system, but instead to act as a companion component that facilitates interoperability with data sharing partners.

RLUS can accommodate both automated and user-modulated request response scenarios. It handles each the same way – through a service-level access to coarse-grained representations of underlying resources, registries, and repositories. As an example, take a simple request for a patient's historical data from Organization A in Figure 1 (above). The user request goes to RLUS, which returns metadata sets that correspond to underlying system repositories. In this case, RLUS serves as a simplifying resource for the organization, as it provides a single point of access for all resource registries.

However, Organization A's RLUS must be able to accommodate an automated scenario as well. In Figure 2, a cross-organizational RLUS functions as a gateway to both

Organization A and B's RLUS-compliant systems. A common scenario may be that a patient's Master Problem List might be automatically generated from information in Organization A and B due to a user's request.

RLUS, then, is an important infrastructure component, but it is not sufficient by itself to accomplish resource location, access, and updating services. Functionally, it relies on components to normalize entities across domains and to make semantic signifiers available during a transaction. Practically, it likely relies on some security implementation as well.

8 Relationship to Information Content

The following principles shall be followed for specifying the information model to be used by the services being specified in this Service Functional Model:

1. SFMs shall provide a conformance profile supporting HL7 content where relevant
2. We shall not preclude the use of non-HL7 content
3. SFMs will reuse to the maximum extent possible the content models as defined in other standards (for example, HL7 RMIMs)
4. Information content representations shall be represented in platform-agnostic formalisms (e.g., UML)
5. SFMs may identify content at varying levels of granularity, depending upon the functions being specified. (For example, the Common Terminology Service will deal with different granularity of information than the Retrieve, Locate, and Update Service).
6. Conformance Profiles may be balloted or adopted after the release of the initial SFM to address specialized business needs. (realm-specific profiles, domain-specific profiles, etc.)
7. Details about semantics specific to this SFM appear in other sections of this document

9 Recommendations for Technical RFP Issuance

This section includes Identification of topics requiring elaboration in candidate solutions provided through the OMG RFP process. These may be service-specific, deployment related, or non-functional.

9.1 Semantic Signifiers

Defining the semantics of payloads sent through RLUS is beyond the scope of this publication. Nevertheless, the ability of RLUS to notify a service partner about the nature of the “what” that is being requested is essential to fulfilling the intent of the request.

RLUS could conceivably be used to indicate the location of and provide access to scanned forms, journal articles, or DICOM images, to name a few. Information sharing partners must keep in mind the notion semantics that are narrowly defined for a particular transaction or set of transactions do not meet the intent of an automated process. To create true interoperability between organizations, it is essential to allow a scalable and extensible model of semantic definition to be included in the description, access, and retrieval of a resource.

Though a limited number of semantic signifiers have been included in this document, it is expected that HL7, HL7 member organizations, professional societies, and localities will be producing these representations that will be supported within RLUS implementations. It is reasonable to expect subsequent adopted standards or publicly available semantic signifier specifications to address specialty needs.

9.1.1 Description of Semantic Signifiers

Where the content exposed through RLUS is from an HL7 domain, describing the semantic signifiers using LIMs is necessary, but may not be sufficient. RFP submitters should take the requirement for LIM description as a starting point to discuss the additional need physical information descriptions. The usage of the two should be described and modeled so as to paint a complete picture of the issue of semantic description and discovery through the RLUS interface.

Additionally, Semantic Signifiers should allow for the use of some sort of logical operators in describing their hierarchy or aggregation. For example, Boolean Operators (AND, OR, NOT) should be available in creating query parameters. This should be discussed in detail by RFP Submitters.

9.2 Service to Service Models

It is tempting to allow RLUS to simply expose existing resource registries within a health organization and to re-create existing business cases using the new technology. This is a completely appropriate pattern for RLUS to follow, but is not necessarily the intent of the functional model.

Services must be able to interoperate freely to support fully automated transactions without any manual intervention or decision making. This means that nearly all aspects of inter-organizational interoperability must be supported by the HSSP specifications. RLUS abides by this overarching principle in being able to accommodate not only business cases that explicitly involve human interaction, but also service to service (automated) business cases. One example is the migration of existing location resources to an RLUS-compliant system, provisioning it with enough information to be operable. Another is the potential to cache certain medical records for individuals for the sake of availability. This process might happen before the patient is actually involved with a human practitioner in the medical workflow. Ultimately, information exchanged by these automated events needs to be as reliable, authoritative, accessible, and usable as any exchange involving human interaction.

In short, as RLUS functionality is realized in implementations that act at the service layer, there is the potential that the existing methodologies of location and retrieval will expand to fit an inter-organizational setting. RLUS must take in to account the ever increasing need for systems to work together to provide a more robust and usable model for healthcare.

9.3 Conformance Profiles and Service Level Agreements

The capabilities defined within this functional model have been attributed to different "conformance profiles." The purpose of conformance profiles is to group together functions to form cohesive "levels" of operational capability against which implementations can be tested for conformance. Thus, interoperability between RLUS implementations is assured within a conformance profile. In other words, two RLUS implementations that conform to the Retrieval profile will be able to interoperate using the functions described in that profile.

These profiles serve to educate the purchasing and implementation communities, allowing for implementation variation while still promoting interoperability. Service Level Agreements made between organizations are then testable because they are informed by these profiles. Governance of these agreements is less ambiguous and more enforceable due to precise functional levels of interoperability that may be expected.

Implementation of this functional specification should explicitly deal with the different interoperability roles that RLUS may fill using these conformance profiles. The business rules enforced by an organization's purchasing, implementation, and governance arms should be discussed, and the ways in which RLUS facilitates that enforcement should be made clear. Additionally, an algorithm for conformance testing should be enumerated, and the means of testing and interpretation of test results should be discussed.

9.4 A special note regarding Resource Retrieval and Information Integration

RLUS conceptually encompasses two information integration models, both direct and indirect brokerage of information. RLUS (Location) would serve well as an indirect information broker as it explicitly does not handle the actual exchange of a resource. RLUS (Retrieval) assumes that RLUS is acting as a direct information broker, managing the exchange of the resource from some server to some client.

Service layers are not necessarily the most efficient carriers for large chunks of data. Conceptually, it is completely appropriate for a service to be RLUS (Retrieval) compliant, but to pass pointers to resources. This could be implemented for security reasons, or it could be implemented due to performance concerns. The distinction will be made on whether a resource is made available explicitly due to a transaction request from RLUS or whether the act of accessing a resource is completely independent of the RLUS mechanisms. In the former case, an implementation of RLUS would be conformant with RLUS (Retrieval) even if the resource was not delivered via RLUS. In the latter case, the implementation could only be said to be compliant with RLUS (Location).

All business cases within this functional model are written with the assumption that RLUS is acting as a direct information broker and that the resource is explicitly passing through RLUS as the transaction is fulfilled. This may not be an accurate technical representation. Therefore, if an RFP Submitter wishes to implement an indirect broker scenario, they should take into account the need for RLUS (Location) and RLUS (Retrieval).

9.5 Operationalizing RLUS: Considerations in Implementation

9.5.1 Automating RLUS

Usage scenarios that do not involve human interaction are expected to be supported by RLUS implementers. This means that certain operational components of RLUS need to be present that are not provided for within the functional specification.

An example is the notion of an "Is RLUS Operational?" interface. In a service to service scenario, it will be important for a consuming service to cope with an RLUS instance that is temporarily disabled because of maintenance or more drastic reasons.

Such features that make RLUS provide for operational robustness should be highlighted and made apparent.

9.5.2 Error Conditions

Because RLUS occupies the service space, the way in which it handles errors needs to be highlighted within any RFP response.

9.5.3 Passing objects: By Reference or By Value

RLUS provides one means of diminishing boundaries between organizations in that it provides for usage scenarios that are intrusive. Any implementation of the RLUS (Updating) conformance profile should call out the way with which object integrity is dealt. Are objects locked, a reference passed, and then changes made to the original? Or is the original made available?

9.5.4 Localization

As a standard, RLUS intends to provide for the broadest acceptance possible while still providing for essential capability. This specification recognizes that essential is not the same as sufficient, especially within healthcare information systems. RFP Submitters should discuss what localization occurred in the following areas: metadata, interfaces, and business scenarios.

9.5.5 Query Parameters

In many cases, the interactions that are scoped for RLUS are exceedingly rich, especially where semantics and semantic signifiers are concerned. Of interest are ways to define not only the outputs of functions and interfaces, but ways to request rich content.

RFP Submitters should discuss ways to extend RLUS to allow complex semantic structures to be passed that contain search parameters and output parameters.

9.5.6 Atomicity of Data

Since resources available through RLUS are described by a semantic signifier, there exists a loose coupling between the form a resource takes as it is expressed through RLUS and its representation within a deployment context. It is essential that RFP Submitters discuss, and if necessary, prescript an appropriate methodology for delimiting the atomicity of data that is available through a particular RLUS instance.

9.5.7 Identification of Resources

As there is a loose coupling between a resource which is exposed through an RLUS interface, it is important to understand the relation between the encapsulation of resources (database versus document) and their identification. The various aspects of this issue should be discussed in detail by implementers, including: identification of aggregated resources, identification of resources across organizational boundaries (within regional *Affinity Domains,* for example), identification of encapsulated resources (documents, for example), and other examples that may come to light.

9.5.8 Data Models and the Realization of Services

It is implicit in much of this document that the realization of RLUS-compliant services, along with the supported interoperability and functionality, is dependent on semantic interoperability. Semantic interoperability begins and ends with trading partners having well defined and accepted data models for their information, regardless of the business need being supported by that information. Thus, RFP submitters will need to require that information being made available through RLUS be modeled and that that model be able to be offered through the semantic interfaces available as part of the RLUS specification. These data models realized as semantic signifiers are expressly supported by the semantic functionality present in this specification.

9.6 Internationalization

What effect, if any, will the localization and internationalization have on technical implementations of RLUS?

9.7 Federation and Cross-Organizational RLUS

RLUS is intended to be a “light-weight” set of components that explicitly occupy the service layer of information technology architectures. It strives to selectively expose certain underlying components and models that align with business needs that arise between organizations. The intention of this specification is to standardize these interfaces according to conformance profiles, allowing for achievable interoperability between trading partners’ information systems. With this intention in mind, the related issues of federation, federation topologies, and cross-organizational implementations become important. Dependencies related to these deployment scenarios deserve special mention.

9.7.1 Service Description and Discovery

Because it exists as a service within a more volatile environment “between” organizations, RLUS should be considered a perfect candidate to benefit from service description and discovery. RFP Submitters should explicitly discuss this deployment case, how to make RLUS less brittle to environmental changes, and how service discovery can possibly be leveraged with regard to the secondary business scenarios described herein.

9.7.2 Federation Topologies

As implementers strive to organize RLUS within and between institutions, the service interfaces will occupy various information and domain levels within an organization, which may or may not map to differing security parameters. Common federation patterns are likely to emerge, such as a mesh or a hierarchical structure.

RFP Submitters should allow for a hierarchical topology to satisfy most deployment requirements. However, other deployment scenarios are desirable as well. Special attention should be paid to implementation in a non-homogeneous security environment.

See also 9.7.3 – RLUS to RLUS Forwarding, below.

9.7.3 RLUS to RLUS Forwarding

In any number of deployment scenarios, multiple RLUS interface sets will likely coexist. Essential to seamless interoperability and simplification of deployment will be the notion of homogenized RLUS deployments in a heterogeneous environment. One tenet of this will be the notion that RLUS can forward requests to other RLUS instances. This functionality is necessary, but not sufficient, to accommodate most federation scenarios.

Such functionality would allow for RLUS deployments to take any form that is deemed expedient as deployments become interchangeable within a topology. RFP Submitters should pay special attention to the issue of query forwarding, and explain how this functionality enables or disables federation topologies and cross-organizational deployments.

10 Glossary

Citation of terms specific to this functional specification and not included in the overall HSSP Glossary

Asset	A informational resource that an organization chooses to make available through RLUS.
Caregiver	Refers to any person, group of people, or organization that imparts health care services to living subjects. This term is intended to neutral and interpreted as reasonable and appropriate in any given implementation realm. Realm-specific interpretation examples include, but are not limited to: In the

	United States, “caregiver” in the context of this specification would commonly be interpreted as a physician or other licensed medical practitioner. In a broader interpretation of this specification, “caregiver” could also include a wide range of ancillary health care personnel such as physicians assistants, nurses, pharmacists, radiology technicians, medical educators, and dietitians, to name a few.
Component	As in Service Component. A component is a modular unit with well-defined interfaces that is replaceable within its environment. A component can always be considered an autonomous unit within a system or subsystem.
Cross-Domain RLUS	See Cross-Organizational RLUS. These are equivalent for the purposes of this specification.
Cross-Organizational RLUS	An inter-organization implementation of RLUS. A cross-organizational RLUS has a child relationship with several parent organizations. For example, it may be informed by one or more local RLUS implementations, utilize security implementations that are regional in nature, normalize entities (patients, providers) across organizations, and use prefixes to define resource identities at each parent organization.
Direct Information Broker	See *Information Broker*. A direct information broker not only makes information available, it also handles the retrieval of the resource. RLUS (Retrieval) would probably be a direct information broker.
Healthcare Practitioner	Term used in the UK with similar scope to the broader meaning of Caregiver as above.
Indirect Information Broker	See *Information Broker*. An indirect information broker makes information available about a resource, but leaves the retrieval of that object to another system. RLUS (Location) might be part of an indirect information broker.
Information Broker	A system that mediates information exchanges. There are generally two aspects of brokering: location and fulfillment. Location would show that a resource is available. Fulfillment would retrieve a chosen resource, for example, or manage a transaction involving that resource. A good example is

	a stock broker, who firsts identifies securities that may be exchanged, and then fulfills a request by buying or selling these securities, completing any legal components involved with transfer of ownership, and notifying the customer that the transaction has been completed.
Localized Information Model (LIM)	LIMs allow additional constraints of information models without impacting the interoperability of instances with other instances conformant to the (more generic) parent Constrained Information Model (CIM). LIMs allow for semantic signifiers to be shared between interoperable instances of services, and are mandated as the mechanism for sharing semantic signifiers when one or more of the service instances is RLUS compliant. LIMs are, by definition, serializable and expressable through UML.
Registry	A registry is understood to be a source of information regarding the identification and location of actual data. A telephone book is an example of a registry.
Repository	A repository is understood to be a place where electronic data is stored. It could be a relational structure, an object-oriented structure, or a file system, depending on the deployment context.
Resource	Any asset, information, data, or object that is modellable and which serves a business requirement by being exposed through an RLUS-compliant interface. The different conformance profiles allow an organization to choose whether the object itself or a representation is made available through the service. Different semantic signifiers allow similar functionality to be extended to different models of information in a consistent and comprehensible manner. Semantic profiles aggregate semantic signifiers in order to describe the information available through the service. Therefore, the scope of information is describable explicitly.
RLUS	Retrieve, Locate, and Update Service
RLUS Entry	An entry in RLUS allows RLUS to reference content, information, resources, or other registries. It is tempting to consider an RLUS entry to equate with a registry entry; this is a convenient model to think of when considering

	RLUS. This notion would be an implementation decision, however. RLUS could take several forms, including being an RLUS of RLUS’s, a file system directory, or any other cataloging construction that contains models of information and the functional means to access those models.
RLUS System	Any record locator implementation that uses the RLUS interface specification.
Semantic Signifier	Semantic Signifiers are identifiers of information constructs that specify the structure and meaning of data. See Section 4.3.2 for an extended description of semantic signifiers.
Semantic Transaction	A semantic transaction is representative of an RLUS transaction that utilizes locally or globally defined semantic signifiers to fulfill the request for a particular resource. Though the scenario is representative, it does describe specifically the idea that a resource is retrieved and semantically transfigured according to some published standard.
Service Level Agreement(SLA)	An agreement between a service consumer and a service provider regarding which services will be delivered and the measurable levels of those services the provider is expected to achieve.
Steward Organization	Each resource is owned, endorsed, and/or managed by a particular organization. Each of these acts constitutes stewardship for the purposes of this specification. None of the concepts are mandatory as RLUS must fit into different scenarios where stewardship is defined differently.

11 Appendix A: Relevant Standards

Review of potentially relevant standards, including a short-list of applicable standards.

For each applicable standard (this may include citations to standards themselves, information content, portions of standards, etc. Demonstrate that “you are not re-inventing the wheel”):

A short review that explains its intended relationship to this specification

What are the relevant parts that are being re-used, extended, etc.

Include context of how the service relates to the existing standard.

How does this work relate to similar work;

What are the implications if this service is used in an environment that has already adopted a competing or closely related standard

If there is relevant realm work, a traceability matrix would be useful here {for instance, U.S. Federal Enterprise Architecture/Service Reference Model}

Applicable Standard	**Relationship**	**Notes**
HL7 CDA	The HL7 CDA Specification may be used as a structure for the payload definition of RLUS-retrieved results. In other words, the parameters on the RLUS service interface may use CDA-conformant representations as the structure and semantic of the data it is managing.	HL7
HL7 HER	See Appendix B.	HL7
HL7 Version 3 Reference Information Model (HL7 V3 RIM)	The HL7 Reference Information Model provides the underpinning for the information semantics that are used in HL7-conformance profiles of the RLUS specification. For these profiles, the RIM and other RIM-derived information models identify the data elements, data types, structure, and underlying terminologies for payload crossing the RLUS service interfaces.	HL7
IHE XDS Cross Enterprise Document Exchange	*Relevant External Work* – In its early versions, the IHE XDS was an example of an RLUS compliant interface that conforms to an RLUS HL7 CDA Location and Retrieval Profile. In practice, both RLUS and the XDS standards assume more generalized standards for information and data exchange. As both standards evolve, their harmonization will become an ongoing effort for both communities. Also see information for the IHE IT Infrastructure Technical Framework.	PDF

Dublin Core Metadata Standards	*Relevant External Work* - The DC Metadata standards provide a common model for content description. It was decided that it would be too ambiguous for this specification because it lacks certain common metadata components, or it would require a complicated domain mapping to arrive at a Dublin Core compliant set of metadata. Nevertheless, the DC gave the SFM the beginning components of metadata that led to the current set.	Dublincore.org
Localized Information Model (LIM)	*Relevant External Work* - The HL7 Organizations Template Special Interest Group has undertaken to provide for description and localization of information models. LIMs provide a way to communicate the informational semantics of an RLUS instance to trading partners. See HL7 Templates below.	HL7
Universal Description, Discovery, and Integration	*Relevant External Work* - UDDI provides a platform for Discovery and Description of Services, and helped to broadly define the business needs for true automated service-to-service RLUS interactions. The UDDI specification informs the RLUS SFM in its notion of topologies and in its design for automated discovery and description. Thus, it defined appropriate functional boundaries and expectations without creating a normative concept. This specification will likely take on added importance in the OMG process.	Uddi.org
World Wide Web Consortium's Universal Resource Identifier (URI)	The W3C's URI specification is important in identifying objects on the Internet, and contains a thorough treatment of the underlying issues of identification and location. It is related to, but not the same as, the IETF's scheme of Universal Resource Naming. URI's deprecates the popular concept of Universal Resource Locations (URL's), in that a URL is an informal concept within URI's. Specifically, URL's identify a resource using its primary access mechanism (Hypertext Transfer Protocol (http)).	W3C
HL7 Templates	The HL7 Templates Special Interest Group (Templates SIG) is presently in the process of harmonizing requirements from among the CEN, OpenEHR, and HL7 communities. Each of these communities is using some form of structure, constraint, and semantic to do precisely the types of representations and uses that are expected of RLUS semantic signifiers.	HL7

HSSP SSDF	The HSSP specifies a Service Specification Development Framework that encompasses the process whereby standards are achieved. This framework also defines the optimal paths for conformance profiling, including functional and semantic roadmaps.	http://hssp.wikispaces.com

Appendix B: HL7 EHR Functional Model Traceability

The functionality defined in the EHR-S model doesn't have a one-to-one exact mapping into the functionality defined by the RLUS service. The functionality provided by RLUS can be seen as basic infrastructure that supports the implementation of EHR-S functionality. For example, function DC.1.1.3.2 Manage medication list needs to access recently updated medication lists in other organizations as well as pharmacy dispense records in order to form a sufficient base for the current medication list. The RLUS doesn't fully implement the function DC.1.1.3.2, but it can be used to implement parts of that function.

EHR Function ID	EHR Function Name	EHR Function Statement	RLUS Interface	Notes *For every row, explain the rationale for including in this specification.*
DC.1.1.3	Manage summary lists	Create and maintain patient-specific summary lists that are structured and coded where appropriate.		Patient summary lists can be created from patient specific data and displayed and maintained in a summary format. The functions below are important, but do not exhaust the possibilities.
DC.1.1.3.1	Manage problem list	Create and maintain patient-specific problem lists.		A problem list may include, but is not limited to: Chronic conditions, diagnoses, or symptoms, functional limitations, visit or stay-specific conditions, diagnoses, or symptoms. Problem lists are managed over time, whether over the course of a visit or stay or the life of a patient, allowing documentation of historical information and tracking the

				changing character of problem(s) and their priority. All pertinent dates, include date noted or diagnosed, dates of any changes in problem specification or prioritization, and date of resolution are stored. This might include time stamps, where useful and appropriate. The entire problem history for any problem in the list is viewable.
DC.1.1.3.2	Manage medication list	Create and maintain patient-specific medication lists.		Medication lists are managed over time, whether over the course of a visit or stay, or the lifetime of a patient. All pertinent dates, including medication start, modification, and end dates are stored. The entire medication history for any medication, including alternative supplements and herbal medications, is viewable. Medication lists are not limited to medication orders recorded by a caregiver, but may include, for example, pharmacy dispense/supply records and patient-reported medications.
DC.1.1.3.3	Manage allergy and adverse reaction list	Create and maintain patient-specific allergy and adverse reaction lists.		Allergens, including immunizations, and substances are identified and coded (whenever possible) and the list is managed over time. All pertinent dates, including patient-reported events, are stored and the description of the patient allergy and adverse reaction is modifiable over time. The entire allergy history, including reaction, for any allergen is viewable. The list(s) include drug reactions that are not

				classifiable as a true allergy and intolerances to dietary or environmental triggers. Notations indicating whether item is patient reported and/or caregiver verified are supported.
DC.1.1.5	Summarize health record	Present a chronological, filterable, and comprehensive review of a patient's EHR, which may be summarized, subject to privacy and confidentiality requirements.		A key feature of an electronic health record is its ability to present, summarize, filter, and facilitate searching through the large amounts of data collected during the provision of patient care. Much of this data is date or date-range specific and should be presented chronologically. Local confidentiality rules that prohibit certain users from accessing certain patient information must be supported.
DC.1.1.6	Manage clinical documents and notes	Create, addend, correct, authenticate and close, as needed, transcribed or directly-entered clinical documentation and notes.		Clinical documents and notes may be created in a narrative form, which may be based on a template. The documents may also be structured documents that result in the capture of coded data. Each of these forms of clinical documentation are important and appropriate for different users and situations.
DC.1.1.7	Capture external clinical documents	Incorporate clinical documentation from external sources.		Mechanisms for incorporating external clinical documentation (including identification of source) such as image documents and other clinically relevant data are available. Data incorporated through these mechanisms is presented alongside locally captured documentation and notes wherever appropriate.

DC.1.1.8	Capture patient-originated data	Capture and explicitly label patient-provided and patient-entered clinical data, and support provider authentication for inclusion in patient history		It is critically important to be able to distinguish patient-provided and patient-entered data from clinically authenticated data. Patients may provide data for entry into the health record or be given a mechanism for entering this data directly. Patient-entered data intended for use by caregivers will be available for their use.
DC.1.5.1	Manage consents and authorizations	Create, maintain, and verify patient treatment decisions in the form of consents and authorizations when required.		Treatment decisions are documented and include the extent of information, verification levels and exposition of treatment options. This documentation helps ensure that decisions made at the discretion of the patient, family, or other responsible party govern the actual care that is delivered or withheld.
DC.1.5.2	Manage patient advanced directives	Capture, maintain and provide access to patient advance directives.		Patient advance directives and caregiver DNR orders can be captured as well as the date and circumstances under which the directives were received, and the location of any paper records of advance directives as appropriate.
DC.3.2.2	Pharmacy communication	Provide features to enable secure bidirectional communication of information electronically between practitioners and pharmacies or between practitioner and intended recipient		When a medication is prescribed, routed to the pharmacy or another intended recipient of pharmacy orders. This information is used to avoid transcription errors and facilitate detection of potential adverse reactions. Upon filling the prescription, information is sent back to the practitioner to indicate that the patient received the medication. If there is a question from the pharmacy, that

		of pharmacy orders.		communication can be presented to the caregiver with their other tasks.
DC.3.2.5	Communication with medical devices	Support communication and presentation of data captured from medical devices.		Communication with medical devices is supported as appropriate to the care setting. Examples include: vital signs/pulse-oximeter, anesthesia machines, home diagnostic devices for chronic disease management, laboratory machines, bar coded artifacts (medicine, immunizations, demographics, history, and identification).

S.2.2	Report generation	Provide report generation features for the generation of standard and ad hoc reports.		A user can create standard and ad hoc reports for clinical, administrative, and financial decision-making, and for patient use - including structured data and/or unstructured text from the patient's health record. Reports may be linked with financial and other external data sources (i.e. data external to the entity). Such reports may include patient-level reports, caregiver/facility/delivery system-level reports, population-level reports, and reports to public health agencies. Examples of patient-level reports include: administratively required patient assessment forms, admission/transfer/discharge reports, operative and procedure reports, consultation reports, and drug profiles. Examples of population-level reports include: reports on the effectiveness of clinical pathways and other evidence-based practices, tracking completeness of clinical documentation, etcetera. Examples of reports to public health agencies include: vital statistics, reportable diseases, discharge summaries, immunization data including adverse outcomes, cancer data, and other such data necessary to maintain the publics' health (including suspicion of newly emerging infectious disease and non-natural events).

S.2.2.1	Health record output	Allow users to define the records and/or reports that are considered the formal health record for disclosure purposes, and provide a mechanism for both chronological and specified record element output.		Provide hardcopy and electronic output that can fully chronicles the healthcare process, supports selection of specific sections of the health record, and allows healthcare organizations to define the report and/or documents that will comprise the formal health record for disclosure purposes.
S.3.2	Information access for supplemental use	Support extraction, transformation and linkage of information from structured data and unstructured text in the patient's health record for care management, financial, administrative, and public health purposes.		Using data standards and technologies that support interoperability, information access functionalities serve primary and secondary record use and reporting with continuous record availability and access that ensure the integrity of (1) the health record, (2) public health, financial and administrative reporting, and (3) the healthcare delivery process
S.3.3.4	Support of service requests and claims	Support interactions with other systems, applications, and modules to support the creation of health care attachments for submitting additional clinical information in		Automatically retrieves structured data, including lab, imaging and device monitoring data, and unstructured text based on rules or requests for additional clinical information in support of service requests or claims at the appropriate juncture in the encounter workflow

		support of service requests and claims.		
I.2	Health record information and management	Manage EHR information across EHR-S applications by ensuring that clinical information entered by providers is a valid representation of clinical notes; and is accurate and complete according to clinical rules and tracking amendments to clinical document. Ensure that information entered by or on behalf of the patient is accurately represented.		Since EHR information will typically be available on a variety of EHR-S applications, an EHR-S must provide the ability to access, manage and verify accuracy and completeness of EHR information, and provide the ability to audit the use of and access to EHR information.

I.2.1	Data Retention and Availability	Retain, ensure availability, and destroy health record information according to organizational standards. This includes: > Retaining all EHR-S data and clinical documents for the time period designated by policy or legal requirement; >Retaining inbound documents as originally received (unaltered); >Ensuring availability of information for the legally prescribed period of time; and >Providing the ability to destroy EHR data/records in a systematic way according to policy and after the legally prescribed retention period.		Discrete and structured EHR-S data, records and reports must be: > Made available to users in a timely fashion; > Stored and retrieved in a semantically intelligent and useful manner (for example, chronologically, retrospectively per a given disease or event, or in accordance with business requirements, local policies, or legal requirements); > Retained for a legally-proscribed period of time; and >Destroyed in a systematic manner in relation to the applicable retention period. An EHR-S must also allow an organization to identify data/records to be destroyed, and to review and approve destruction before it occurs.

I.2.4	Extraction of health record information	Manage data extraction in accordance with analysis and reporting requirements. The extracted data may require use of more than one application and it may be pre-processed (for example, by being de-identified) before transmission. Data extractions may be used to exchange data and provide reports for primary and ancillary purposes.		An EHR-S enables an authorized user, such as a clinician, to access and aggregate the distributed information, which corresponds to the health record or records that are needed for viewing, reporting, disclosure, etc. An EHR-S must support data extraction operations across the complete data set that constitutes the health record of an individual and provide an output that fully chronicles the healthcare process. Data extractions are used as input to continuity of care records. In addition, data extractions can be used for administrative, financial, research, quality analysis, and public health purposes.
I.3.1	Distributed registry access	Enable system communication with registry services through standardized interfaces and extend to services provided externally to an EHR-S.		An EHR-S relies on a set of infrastructure services, directories, and registries, which may be organized hierarchically or federated, that support communication between EHR-S'. For example, a patient treated by a primary care physician for a chronic condition may become ill while out of town. The new caregiver's EHR-S interrogates a local, regional, or national registry to find the patient's previous records. From the primary care record, a remote EHR-S retrieves relevant information in conformance with applicable patient privacy and confidentiality

				rules. An example of local registry usage is an EHR-S application sending a query message to the Hospital Information System to retrieve a patient's demographic data.

12 Appendix C: The HL7 CDAr1 Document Profile

12.1 Introduction

It is not the intention of this specification to provide an entire CDA implementation guide or to provide a robust semantic profile to meet localized business requirements. Rather, the HL7 CDAr1 Document profile contains a minimalist implementation of the CDA Release 1 specification that is designed to allow a solution to the business needs outlined herein, and to provide a simple foundation that may be extended to meet an organization's needs.

12.2 Semantic Profile

The HL7-CDAr1 Document Profile instances consist of the following:

- Clinical Document Header
    - Description: A required structure to hold essential information about the document in question
    - Cardinality: 1..1
    - Minimum Trait Set:
        - Identifier

        - Description: A required, globally unique instance identifier
        - Cardinality: 1..1
        - Data Type: HL7 v2 Instance Identifier
    - Provider
        - Description: A healthcare provider who participated in the service(s) being documented
        - Cardinality: 1..*
        - Data Type: Complex
            - Hierarchy:
                - provider
                    - type_cd
                    - function_cd
                    - participation_tmr
                    - person
                        - id
                        - person_name
                            - effective_tmr
                            - nm
                            - type_cd
                            - addr
    - Patient
        - Description: The principle subject of the service being documented
        - Cardinality: 1..*
        - Data Type: Complex
            - Hierarchy:
                - type_cd
                - participation_tmr

- person
  - id
  - person_name
    - effective_tmr
    - nm
    - type_cd
    - addr
- is_known_by
  - id
  - is_known_to
    - id
- birth_dttm
- administrative_gender_cd

- Clinical Document Body
  - Description: A required structure to hold the actual payload that is being shared between trading partners. It may be populated with structured elements logically subdivided into sections, or with unstructured data (images, binary data, &c.).
  - Cardinality: 1..1
  - Minimum Trait Set: “Section” AND / OR “Unstructured Data”
    - Section
      - Description: Sections are logical delimiters of a document being processed with the Clinical Document Architecture
      - Cardinality: 1..*
      - Data Type: ANY
    - Unstructured Data
      - Description: A portion of the body where unstructured data may be placed
      - Cardinality: 1..1
      - Data Type: ANY